

---

# WAPT Documentation

*Version 2.1*

**Tranquil-IT Systems**

janv. 09, 2024









Bienvenue sur la documentation officielle de WAPT par Tranquil IT dernière version en date du 2024-01-09.

Cliquer [ici](#) pour une version PDF de la documentation complète.

**WAPT est un outil de déploiement de logiciels et de configurations** qui peut être comparé à Microsoft SCCM (System Center Configuration Management) (maintenant appelé MECM (Microsoft Endpoint Configuration Management)), Ivanti UIM (Unified Endpoint Manager), IBM Bigfix, Tanium, OPSI, PDQDeploy, ou Matrix42. WAPT existe en deux versions, *WAPT Discovery* et *WAPT Enterprise*.

#### **Pour les Administrateurs Système :**

- Installer des logiciels de manière silencieuse.
- Maintenir à jour les logiciels installés et leurs configurations.
- Configurer les logiciels et le système pour diminuer la charge sur les équipes de support.
- Désinstaller de manière silencieuse les logiciels ou configurations obsolètes.
- Réduire le besoin de support par les équipes informatiques, dont les temps de réaction sont parfois long du fait de leurs charges de travail.
- Réduire autant que possible la consommation de bande passante sur les sites distants afin de la préserver pour des usages productifs.

#### **Pour les RSSI**

- Faire converger les logiciels installés vers la norme de sécurité acceptable pour l'entreprise.
- Préparez votre entreprise à l'arrivée du RGPD et aider votre DPD, qui deviendra un proche collègue, à tenir à jour son registre des traitements.
- Ne plus tolérer le fonctionnement des machines en mode *Administrateur*.
- Ne plus tolérer que les utilisateurs téléchargent et exécutent des logiciels à partir de leur répertoire personnel.
- Commencer à appliquer les SRPs (Software Restriction Policies), également connues sous le nom de *Applocker* ou WDAC (Windows Defender Application Control) pour améliorer la sécurité informatique au niveau des applications.
- Réduire le niveau d'exposition aux vulnérabilités des logiciels et aux attaques par [mouvement latéral](#).
- Faire remonter des indicateurs d'audit pour une meilleure connaissance de l'état des équipements informatiques installés et de leur niveau de sécurité global.
- Déployer immédiatement pour réagir à une menace type [Wannacry](#) ou [notPetya](#).

#### **Pour les utilisateurs finaux**

- Avoir installé des logiciels configurés pour bien fonctionner dans le contexte de votre organisation et avoir confiance qu'ils fonctionneront correctement.
- Rendre les *Utilisateurs* plus autonomes pour installer des logiciels de manière sûre et sécurisée.
- Disposer de systèmes professionnels plus performants et plus prévisibles grâce à des configurations logicielles standard.



---

## Présentation des grands principes de WAPT

---

### 1.1 A quoi sert WAPT ?

**WAPT** installe, met à jour et supprime les logiciels et les configurations sur les appareils Windows, Linux et macOS. Le déploiement de logiciels (Firefox, MS Office, etc.) peut être effectué à partir d'un serveur central à l'aide d'une console graphique. WAPT reprend de nombreuses idées de l'outil de gestion de paquets apt Debian Linux, d'où son nom.

Des entreprises privées de toutes tailles, des collèges, des écoles, des universités, des laboratoires de recherche, des gouvernements locaux et nationaux, des hôpitaux, des mairies et des ministères d'État du monde entier utilisent avec succès **WAPT**.

**WAPT** existe en deux versions, **Discovery** et **Enterprise**, toutes deux propriétaires, la version **Community** ayant été amicalement *forkée* à la communauté Opensource.

**WAPT** est très efficace pour répondre aux **besoins récurrents de mise à jour de Firefox ou Chrome** et c'est souvent pour couvrir ce besoin de base que WAPT est initialement adopté ; il devient alors un outil de choix pour les tâches quotidiennes de l'administrateur système.

### 1.2 Certification de sécurité de l'ANSSI

Suite à sa certification CSPN du 14 février 2018, WAPT a obtenu le 15 mars 2018 la [Qualification Élémentaire](#) de l'ANSSI.



FIG. 1 – Visa de sécurité de l'ANSSI du 14 février 2018 pour WAPT Enterprise Edition 1.5.0.13



## 1.3 Genèse WAPT

### 1.3.1 Notre constat après 15 ans d'infogérance

L'administration d'un large parc de PC sous Microsoft Windows est aujourd'hui une tâche difficile dans un environnement sécurisé :

- Les méthodes généralement utilisées (mastérisations type *ghost* ou *clonezilla*) sont efficaces si les parcs machines et les parcs applicatifs sont homogènes et que les profils utilisateurs sont itinérants.
- Les outils de télé-déploiement (*OCSInventory* ou *WPKG*) diffusent les logiciels mais ne permettent pas d'effectuer de manière simple les personnalisations qui évitent les demandes de support utilisateur.
- Les logiciels de petits éditeurs nécessitent souvent des droits *Administrateur Local* pour fonctionner correctement.
- Les solutions actuellement disponibles pour résoudre ces problèmes sont soit trop coûteuses, soit trop inefficaces, et elles sont dans tous les cas trop complexes.

### 1.3.2 Hypothèses et motivations du développement WAPT

Le développement de WAPT est animé par deux principes :

- Ce qui est **compliqué** doit être rendu **simple**.
- Ce qui est **simple** doit être rendu **trivial**.

WAPT s'appuie sur un jeu d'hypothèses fondamentales :

- Les adminsys doivent connaître un langage de script, et WAPT a choisi Python pour la profondeur et l'étendue de ses bibliothèques.
- Les administrateurs système qui ont peu d'expérience avec les langages de script doivent s'inspirer d'exemples simples et efficaces qu'ils sauront adapter à leurs besoins.
- Les adminsys doivent pouvoir communiquer sur l'efficacité de leurs actions à leur direction et reporter les écarts de processus aux auditeurs internes ou externes.
- Les adminsys doivent pouvoir collaborer avec leur équipe informatique ; ainsi les dépôts WAPT internes fournissent des paquets auxquels ils peuvent faire confiance pour les déployer sur leur réseau. Sinon, ils peuvent choisir des dépôts externes publics qui leur fournissent les garanties de sécurité qu'ils jugent suffisantes.
- Les administrateurs système sont conscients que les postes de travail des utilisateurs servent à des fins commerciales et que certaines personnalisations doivent être possibles. L'adaptation de l'infrastructure aux besoins de l'entreprise est facilitée par la notion de groupes et des :abbr :OU (*Organizational Units*) ; ils permettent de sélectionner un grand nombre de machines pour personnaliser leur configuration.



### 2.1 Principe de Dépôt

Les paquets sont stockés dans un répertoire web. Ils ne sont pas stockés dans une base de données.

---

**Note :** Le protocole de transport utilisé pour le déploiement des paquets est le **HTTPS**.

---

Les paquets WAPT sont servis par le serveur web **Nginx**, disponible sous Linux et Windows.

Le fichier d'index Packages est la seule chose nécessaire. Il liste les paquets disponibles sur les dépôts autorisés et quelques informations de base sur chaque paquet.

Ce mécanisme permet de mettre en place facilement un processus de réplication entre plusieurs dépôts.

Les grandes organisations avec des sites distants et des filiales nécessitent parfois que les services soient répliqués localement pour éviter la congestion de la bande passante (*Edge Computing*).

### 2.2 Dépôts répliqués

**WAPT Enterprise offre la possibilité de mettre à niveau les agents distants pour servir de dépôt distants pouvant être gérés directement depuis la console WAPT. Tous les agents WAPT peuvent ensuite être configurés de manière centralisée pour sélectionner automatiquement le meilleur dépôts en fonction d'un ensemble de règles.**

Lorsque WAPT est utilisé sur des sites distants à bande passante limitée, il est logique d'avoir un appareil local qui répliquera le dépôt WAPT principal pour réduire la bande passante réseau consommée lors du déploiement des mises à jour sur vos appareils distants.

Avec les dépôts distants, WAPT reste une solution à faible coût d'exploitation car vous n'avez pas besoin de mettre en place **des liaisons fibre haut débit** pour profiter de WAPT.

Cela fonctionne comme suit :

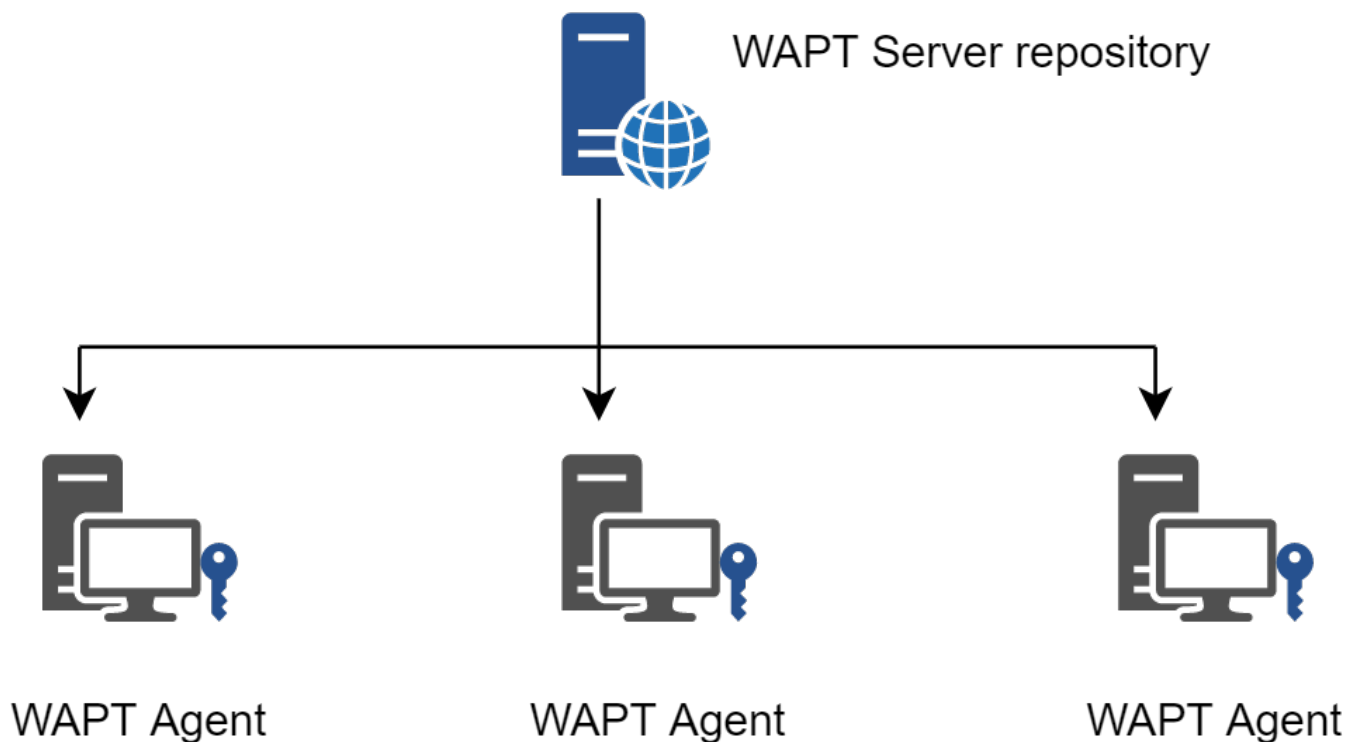


FIG. 1 – Réplication et dépôts multiples

- Une appliance de petite taille et sans maintenance jouant le rôle de dépôt secondaire est déployée sur le réseau local de chaque site distant ; un poste de travail peut également être utilisé, même s'il peut ne pas être opérationnel si vous souhaitez vous y connecter.
  - Le dépôt distant réplique les paquets du dépôts principal.
  - Les agents WAPT se connectent en priorité au dépôts le plus proche d'eux, le dépôt local.
- Pour en savoir plus sur le dépôt répliqué, consultez la documentation sur *Réplication d'un dépôt*.

## 2.3 Principe de Paquets

La structure d'un paquet WAPT est similaire à celle d'un paquet **.deb** de Debian Linux. Chaque paquet WAPT embarque avec lui les binaires qui seront exécutés et les autres fichiers dont il aura besoin.

Un paquet est transportable facilement.

Voici à quoi ressemble un package WAPT :

Pour en savoir plus sur la composition d'un paquet WAPT, consultez la documentation sur la structure détaillée d'un paquet.

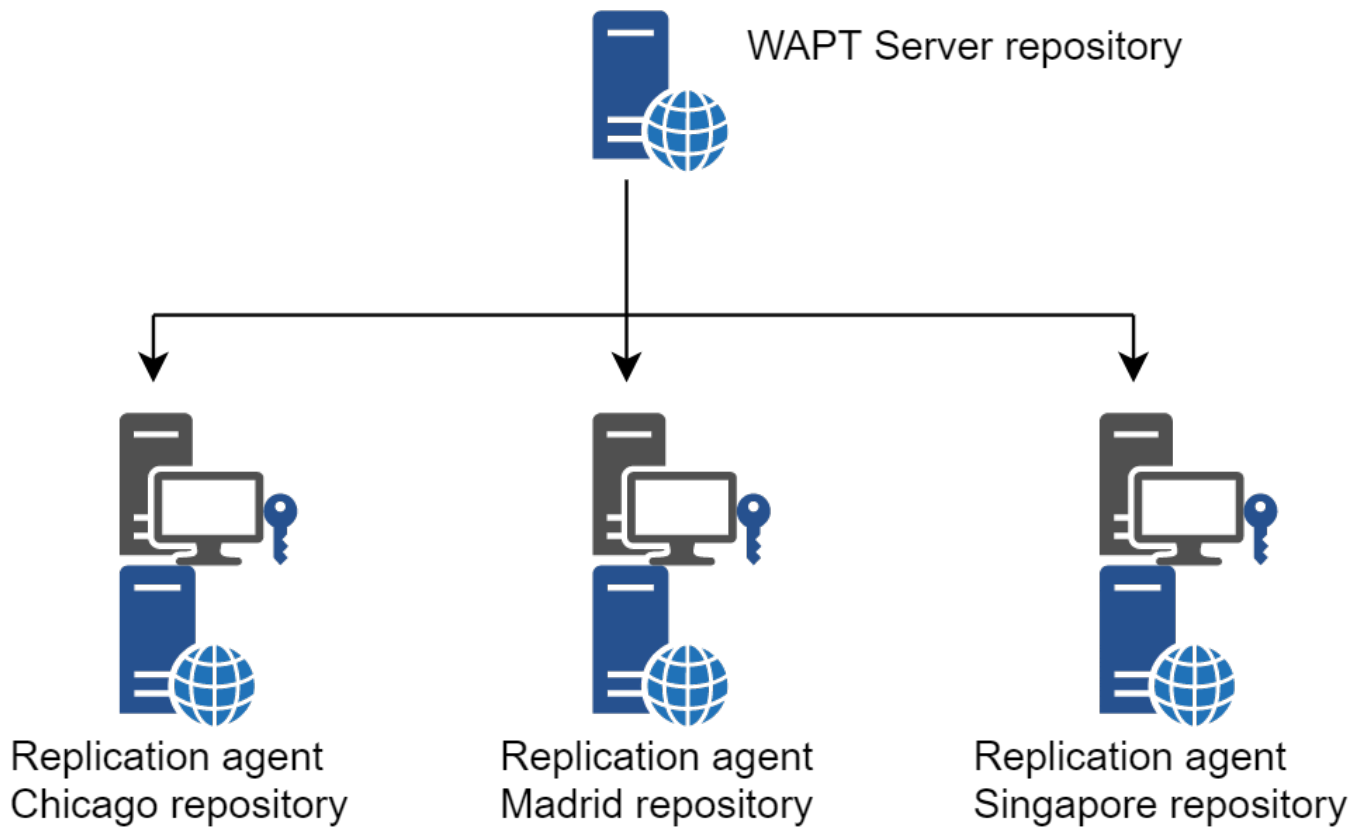


FIG. 2 – Réplication des dépôts WAPT

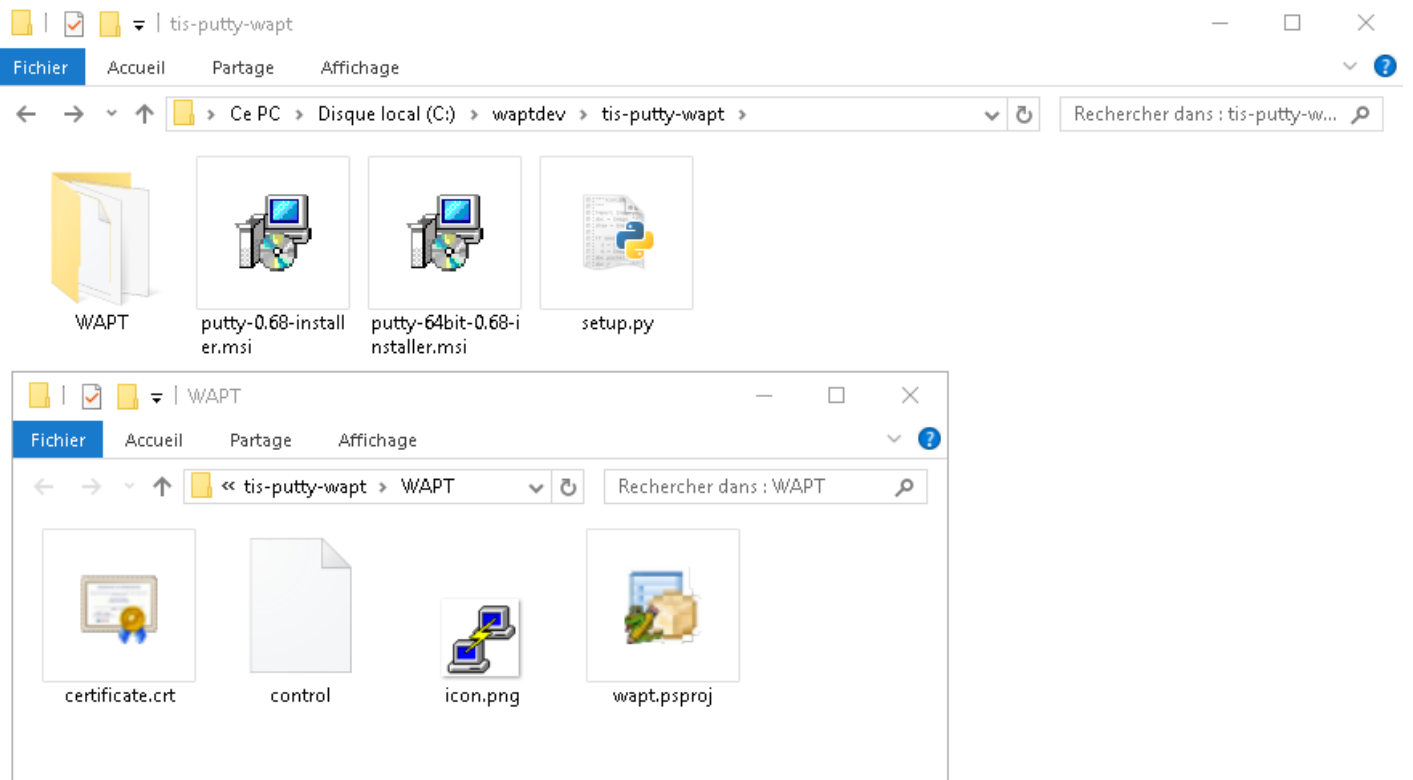


FIG. 3 – Structure d'un paquet WAPT

### 2.3.1 Types de paquets WAPT

Il existe 7 types de paquets WAPT :



FIG. 4 – Représentation d'un paquet WAPT simple

#### Les paquets *base*

Ce sont les paquets logiciels classiques.

Ils sont stockés dans le répertoire web <https://srvwapt.mydomain.lan/wapt/>.

#### Les paquets *group*

Ce sont des groupes de paquets.

Chaque groupe correspond souvent à :

- service dans l'entreprise (ex : **comptabilité**).
- une pièce, un bâtiment, etc.

---

**Indication :** Un client peut-être membre de plusieurs groupes.

---

Ils sont stockés dans le répertoire web <https://srvwapt.mydomain.lan/wapt/>.

#### Les paquets *host*

Les paquets « machines » portent le nom *UUID* Bios ou le *FQDN* de la machine.

Chaque client recherchera son paquet **host** pour connaître les packages qu'il doit installer (*dépendances*).

Les paquets **host** sont stockés dans le répertoire web <https://srvwapt.mydomain.lan/wapt/>.

### les paquets *unit*

Les paquets « *unit* » portent le nom complet d'une OU (Unité Organisationnelle), exemple : **OU=pièce1,OU=prod,OU=computers,DC=mydomain,DC=lan**.

Par défaut, chaque ordinateur recherche les paquets *unit* puis installe la liste des dépendances associées.

Les paquets **Unit** sont stockés dans le répertoire web <https://srvwapt.mydomain.lan/wapt/>.

### les paquets *wsus*

Les paquets *wsus* contiennent le liste des mises à jour Windows autorisées et interdites.

Lorsque ce paquet est installé sur le terminal, la prochaine analyse de mise à jour effectuée par WAPT choisira les mises à jour Windows en fonction de ce filtrage.

Les paquets **wsus** sont stockés dans le répertoire web <https://srvwapt.mydomain.lan/wapt/>.

### les paquets *self-service*

Les paquets *self-service* contiennent une liste de groupes ou d'utilisateurs (Active Directory ou local) et leurs listes associées de paquets que les utilisateurs seront autorisés à installer par eux-mêmes.

Les paquets **self-service** sont stockés dans le répertoire web <https://srvwapt.mydomain.lan/wapt/>.

### les paquets *profile*

Les paquets *profile* sont similaires aux paquets *group*.

Cependant, les paquets *profile* fonctionnent un peu différemment et sont plus utiles lorsqu'un serveur Active Directory fonctionne dans l'*Organisation* :

## 2.4 Principe de dépendance

Dans WAPT tout fonctionne selon le principe de dépendance.

Par défaut, l'agent WAPT recherchera son paquetage *host*. Le paquet *host* liste les paquets à installer sur l'ordinateur.

Ainsi, le paquet *host* sera correctement installé si toutes ses dépendances sont satisfaites.

Chaque sous-dépendance doit être satisfaite pour satisfaire une dépendance de niveau supérieur.

Lorsque toutes les dépendances sont satisfaites, l'hôte notifie son statut au serveur WAPT. Son indicateur devient **OK** et vert dans la console WAPT, ce qui signifie que l'hôte a le profil d'hôte que le *Administrator* ou *Package Deployer* a défini pour lui.

---

**Indication :** Lorsque l'on attribue un logiciel à un hôte en tant que dépendance, seul le nom canonique du logiciel sans son numéro de version est enregistré comme dépendance (ex : Je veux que Freemind soit installé sur cette machine dans sa dernière version et que **Freemind** soit configuré pour que le *User* ne m'appelle pas parce qu'il ne trouve pas l'icône sur son bureau !)

---

Pour chaque dépendance, l'agent WAPT se chargera d'installer automatiquement la dernière version disponible du paquet. Ainsi, si plusieurs versions de **Freemind** sont disponibles sur le dépôt, l'agent WAPT obtiendra toujours la dernière version, à moins que j'aie épinglé la version pour des raisons de compatibilité avec d'autres ensembles d'outils.



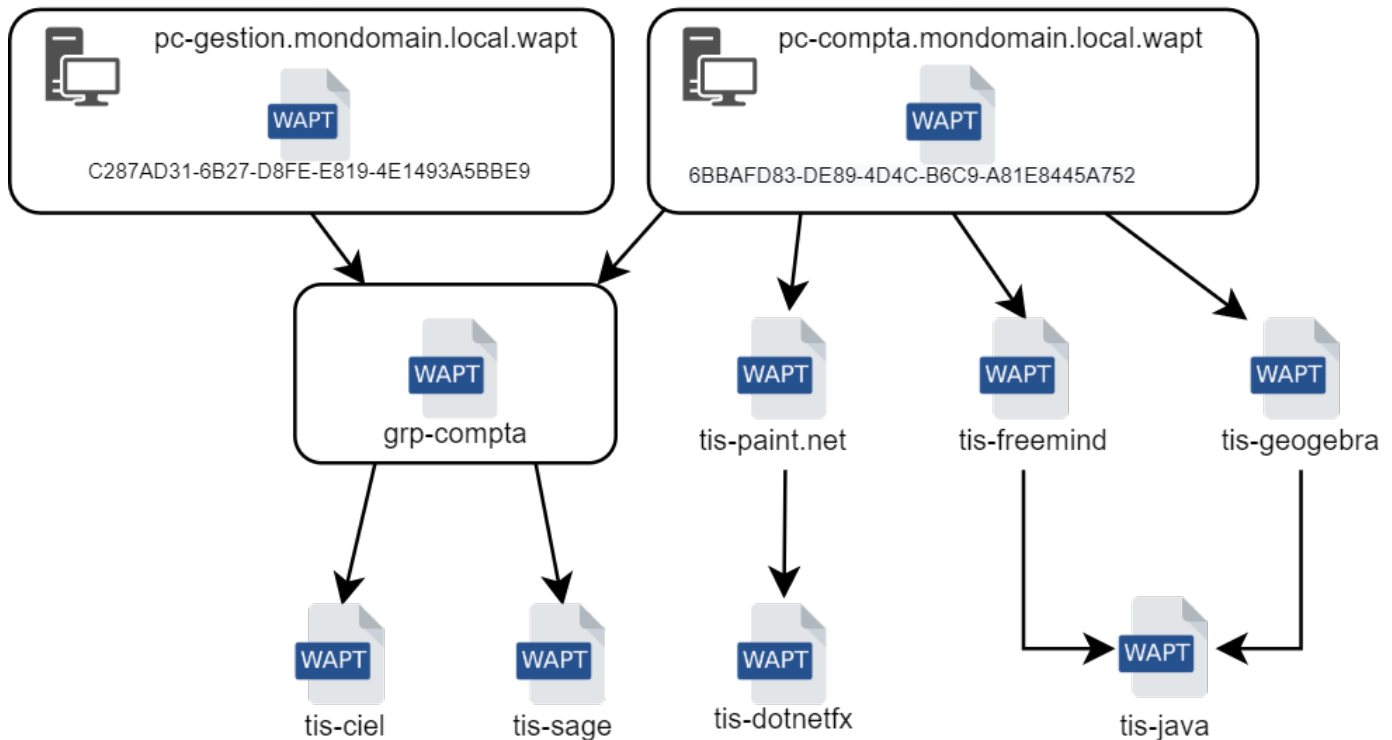


FIG. 5 – Schéma conceptuel du mécanisme de dépendance

Ensuite, lorsque l'agent contacte le dépôt pour vérifier s'il y a de nouvelles mises à jour, il compare les versions des paquets du dépôt avec sa propre liste locale de paquets déjà installés sur la machine.

Si une mise à jour d'un paquet déjà installé est disponible, le client basculera le statut du paquet en **NEED UPGRADE**. Il installera ainsi les mises à jour au prochain **upgrade**.

## 2.5 Principe de Clé privée / certificat public

Comme les paquets Android **APK**, les paquets WAPT sont signés ; un hash de la somme de contrôle de tous les fichiers contenus dans le paquet est calculé.

Cette méthode de signature permet de garantir la provenance et l'intégrité du paquet.

Pour fonctionner correctement, WAPT a besoin d'une paire clé privée /certificat public (auto-signée, émise par une autorité de certification interne *Certificate Authority* ou commerciale).

La **clé privée** sera utilisée pour **signer** les paquets WAPT tandis que le **certificat public** sera distribué avec chaque agent WAPT afin que les agents WAPT puissent valider les fichiers qui ont été signés avec la clé privée.

Les différents certificats publics seront stockés dans le sous-dossier `ssl` de l'agent WAPT. Ce dossier peut contenir plusieurs certificats publics.

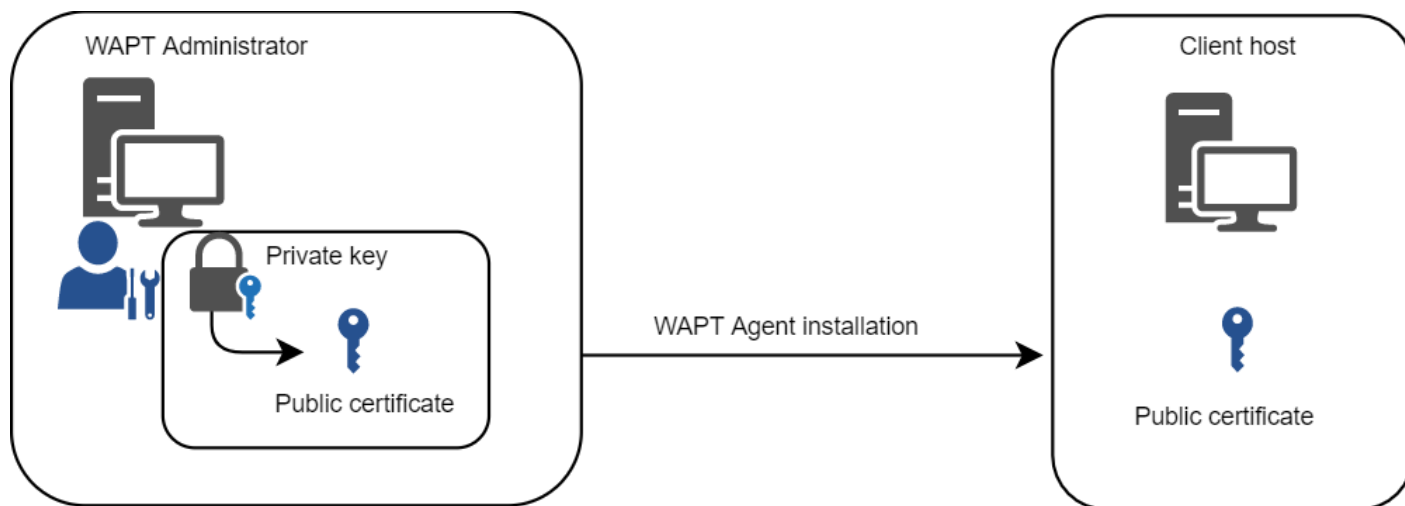


FIG. 6 – Clé privée / certificat public

## 2.5.1 Vérification des paquets

Lorsqu'un paquet WAPT est téléchargé, l'agent WAPT (*waptagent*) vérifie l'intégrité du paquet, puis vérifie que le paquet a été correctement **signé**.

Si la signature du paquet WAPT ne correspond à aucune des certificats publics situés dans `C:\Program Files (x86)\wapt\ssl` sur Windows ou `/opt/wapt/ssl` sur Linux et MacOS, l'agent WAPT refusera d'installer le paquet.

Pour plus d'informations, veuillez consulter la documentation sur *comment l'intégrité du processus d'installation d'un paquet WAPT est assurée*.

## 2.5.2 La clé privée est importante

**Attention :** La clé privée ne doit **PAS** être stockée sur le serveur WAPT, ni sur aucun stockage public ou partagé auquel pourrait accéder du personnel non autorisé. En effet, la sécurité de WAPT repose sur le maintien de la clé privée **privée**.

La clé privée doit être stockée en lieu sûr, car **celui qui contrôle votre clé contrôle votre parc !**

Enfin, pour un maximum de sécurité, la clé privée pourra être sécurisée sur une smartcard ou un jeton cryptographique que les *Administrateurs* et *Déployeur de Paquet* transporteront physiquement sur eux, utilisant leur smartcard ou leur jeton cryptographique ponctuellement pour signer un paquet WAPT.

**Note :** La clé privée est protégée par un mot de passe par défaut.

Plus d'informations sur *générer le certificat de l'administrateur pour signer les paquets WAPT*.

### 2.5.3 Différenciation des rôles des utilisateurs dans WAPT

WAPT offre la possibilité de différencier les rôles en fonction de :

- Une A PKI (Infrastructure à Clé Publique).
- ACL (Access Control Lists).

#### Infrastructure à Clé Publique (PKI)

---

**Indication :** L'utilisation d'une PKI existante est possible, la Console WAPT est livré avec un générateur simple de certificat.

---

WAPT fonctionne comme un mode CA (autorité de certification) en ce qui concerne la PKI.

De par sa conception, WAPT est capable de générer des certificats qui peuvent être utilisés comme clés parent pour générer d'autres clés enfant publiques et privées.

Par conséquent, l'administrateur principal de WAPT qui agit en tant qu'administrateur peut émettre des certificats pour chaque administrateur informatique afin que leurs actions puissent être identifiées lorsqu'ils utilisent WAPT.

Les certificats enfants émis par la CA peuvent eux-mêmes être configurés comme :

- La signature de code pour permettre aux administrateurs informatiques d'emballer, de signer et de déployer des paquets WAPT contenant des charges exécutables (c'est-à-dire `setup.py`).
- CA pour déléguer à d'autres administrateurs informatiques le droit d'émettre des certificats.
- Aucun droit pour limiter les administrateurs informatiques au seul déploiement de paquets contenant des charges non exécutables (c'est-à-dire configurer des hôtes).

Plus d'informations sur *générer l'autorité de certification (CA)*.

#### Liste de contrôle d'accès (ACL)

Avec WAPT, il est possible de définir les droits des utilisateurs en utilisant ACL.

Chaque technicien informatique est identifié par son propre certificat et les droits peuvent donc être appliqués finement sur une base individuelle.

Par exemple, un utilisateur de la console WAPT peut avoir le droit de « Voir » sur une machine mais ne pas être autorisé à cliquer sur « Modifier la machine ».

Plus d'informations sur la *La liste des droits de l'ACL*.

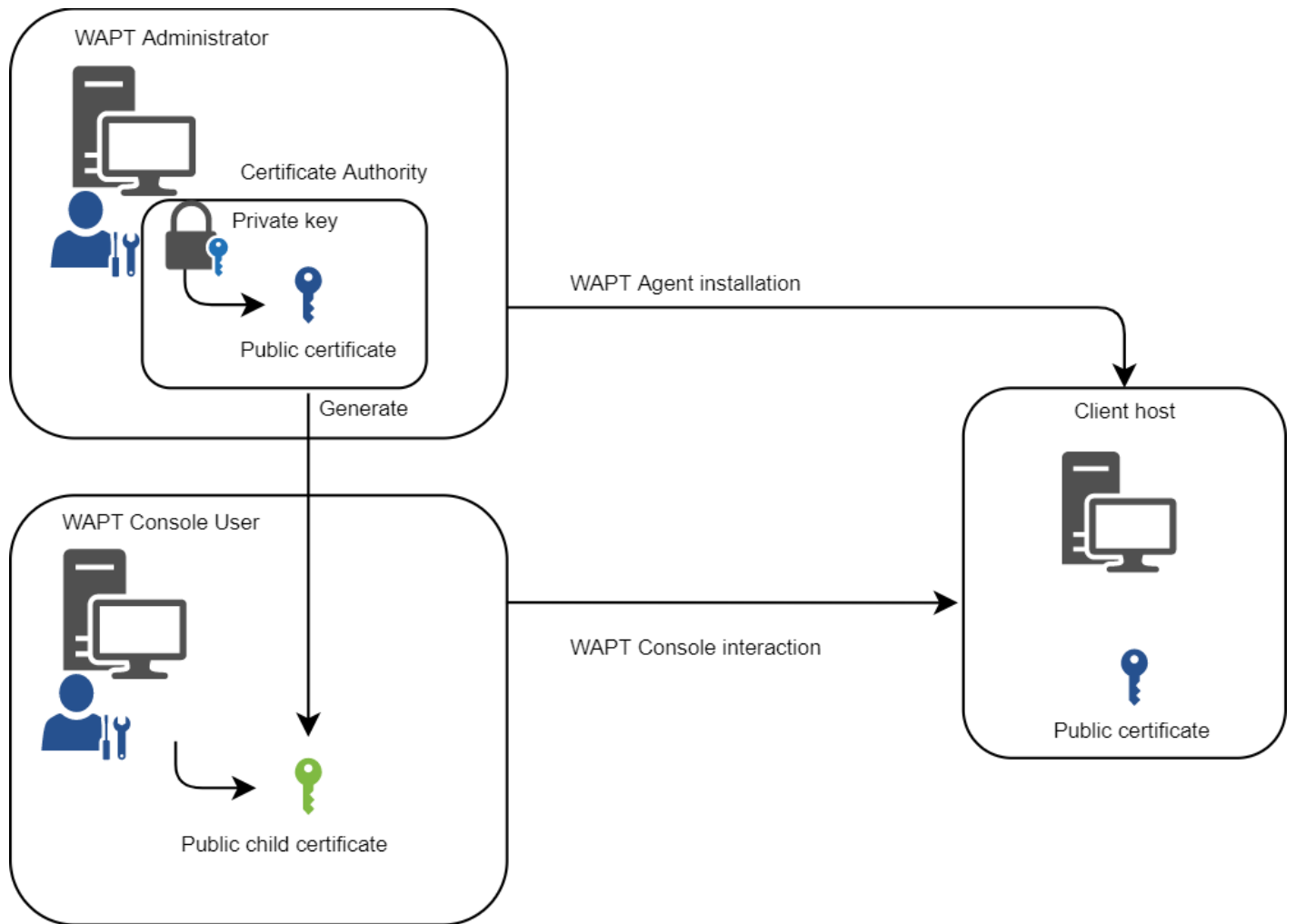


FIG. 7 – Différenciation des rôles des utilisateurs WAPT

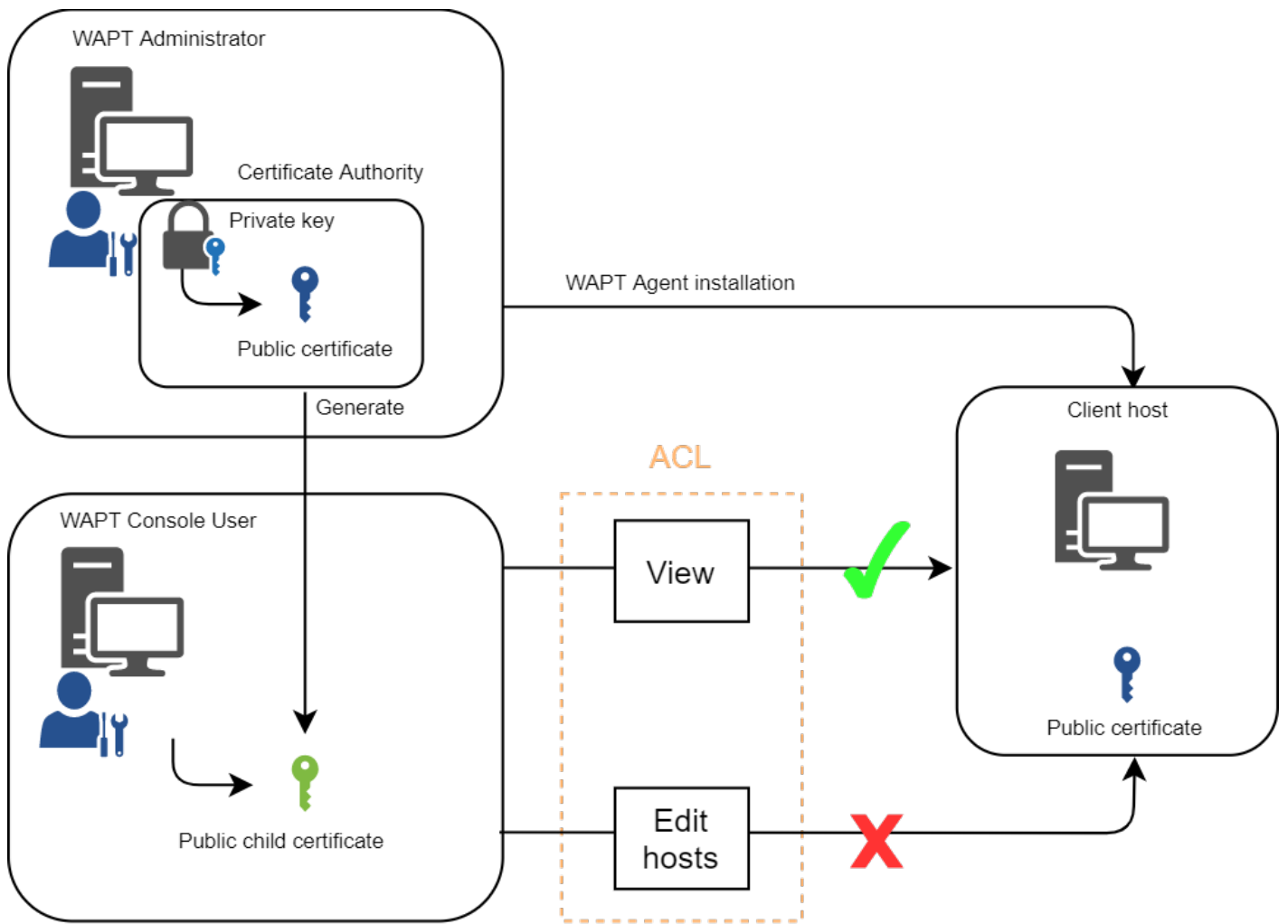


FIG. 8 – Différenciation des rôles ACL



### 3.1 Inventaire logiciel

WAPT tient un inventaire matériel et logiciel de chaque machine.

Cet inventaire est stocké dans une petite base de données intégrée à chaque agent WAPT.

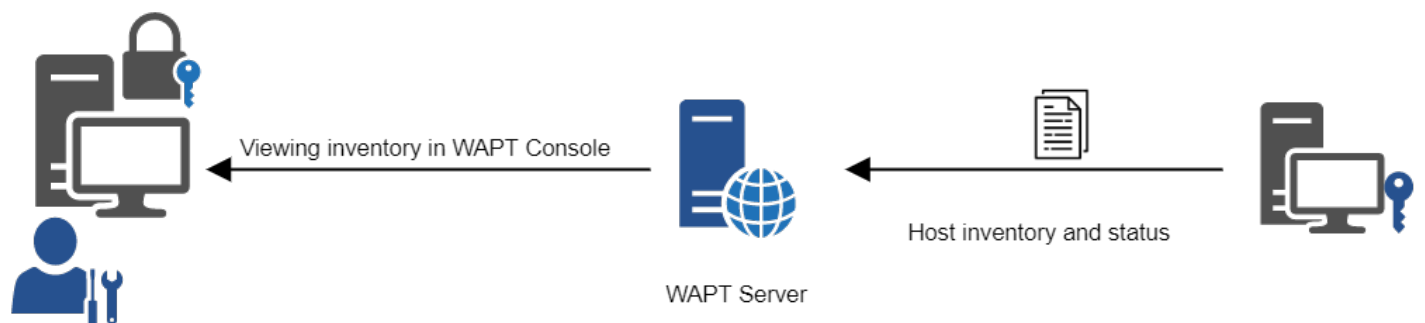


FIG. 1 – Fonctionnement de la remontée d'inventaire

- Lors du premier enregistrement avec le serveur WAPT, l'agent WAPT envoie l'inventaire complet (BIOS, matériel, logiciel) au serveur.
- Lors de chaque mise à jour du client, l'agent WAPT remonte le delta de l'inventaire au serveur.

L'inventaire central vous permet de filtrer les hôtes par leurs composants, leurs logiciels ou tout autre argument de recherche.

Overview Hardware inventory Software inventory Tasks

Filter:  Add item as grid column

Property	Value
+ wmi	
+ dmi	
- host_info	
+ profiles_users	
+ local_administrators	
- mac	
0	42:c3:40:63:7f:c7
system_productname	HVM domU
- connected_ips	
0	192.168.149.149
- local_drives	
+ D	
+ C	
domain_name	null
- current_user	
0	admin
domain_controller	null
wua_agent_version	7.6.7601.23806
virtual_memory	2147352576
computer_ad_site	
- windows_startup_items	
run	
+ common_startup	
system_manufacturer	Xen
description	administrateur demo
computer_ad_dn	
registered_organization	Orgname
win64	True
- networking	
+ 0	
domain_controller_address	null
- windows product infos	

FIG. 2 – L’inventaire dans la console WAPT



## 3.2 La remontée des informations d'inventaire

L'agent WAPT remonte également le statut des paquets WAPT.

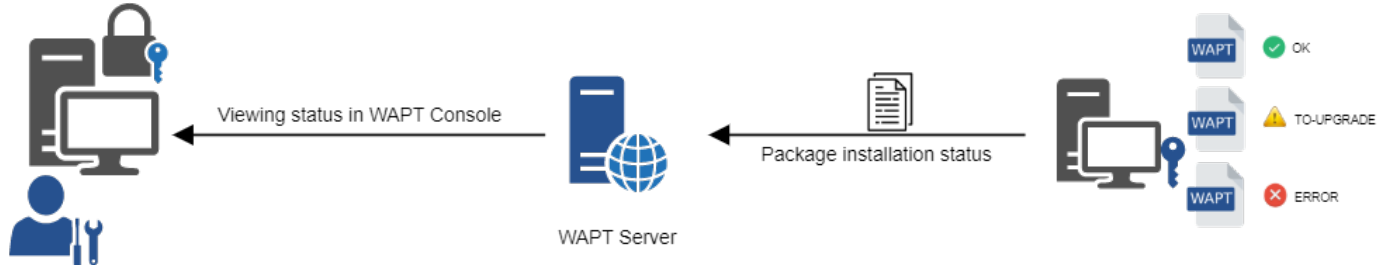


FIG. 3 – La remontée du statut des paquets vers le serveur WAPT

En cas d'erreur lors de l'installation du paquet, l'information sera transmise au serveur WAPT. La machine apparaîtra alors en **ERROR** dans la console.

ERROR	UN...	pc-utilisateur.tra...
ERROR	OK	wsmanage-cog.t...
ERROR	UN...	wm-bma.tranqui...

FIG. 4 – Paquets avec un statut d'erreur dans la console WAPT

Le *Administrator* peut voir le paquet retourné en erreur dans la console et corriger le paquet en conséquence.

Pour chaque **upgrade**, WAPT essaiera d'installer une nouvelle version du paquetage jusqu'à ce qu'aucun statut d'erreur ne soit renvoyé.

---

**Note :** Les agents WAPT signent leur inventaire avant de l'envoyer au serveur WAPT.

Pour plus d'informations, veuillez vous reporter à *Signature des remontées d'inventaire*.

---

## 3.3 Les interactions classiques de WAPT

### 3.3.1 update

Lorsqu'une commande **update** est lancée sur un agent, cela revient à ordonner à l'agent de vérifier le dépôt WAPT pour les nouveaux paquets. **Par défaut, l'agent WAPT recherche les mises à jour toutes les deux heures.**

Si la date du fichier d'index Packages a changé depuis la dernière **update**, alors l'agent WAPT télécharge le nouveau fichier Packages (entre 20 et 100k), sinon, il ne fait rien.

L'agent WAPT compare ensuite le fichier Packages avec sa propre base de données locale.

Si l'agent WAPT détecte qu'un paquet doit être ajouté ou mis à jour, il fait passer le statut de l'hôte et celui du paquet à **NEED-UPGRADE**.

Il ne lancera pas l'installation du paquet immédiatement. L'agent WAPT attendra un ordre « **upgrade** » pour lancer la mise à niveau.

### 3.3.2 upgrade

Lorsque nous lançons une **upgrade**, nous demandons à l'agent WAPT d'installer les paquets ayant un statut *NEED-UPGRADE*.

Une **update** doit précéder une **upgrade**, sinon l'agent ne saura pas si des mises à jour sont disponibles.

Par défaut, l'agent WAPT déclenchera une **update/ download-upgrade** au démarrage ; après le démarrage, l'agent WAPT vérifiera ensuite toutes les 2 heures s'il a quelque chose à faire.

Les paquets à installer seront téléchargés et mis en cache dans le dossier `C:\Program Files (x86)\wapt\cache`.

Le *waptexit* lancera une **upgrade** lorsque l'ordinateur s'éteindra. Un *Administrator* pourra également forcer le lancement immédiat d'une **upgrade** à partir de la console WAPT. Un utilisateur final peut également choisir de lancer manuellement une **upgrade**. Enfin, une tâche planifiée peut être configurée sur les hôtes pour lancer une **upgrade**.

Si le serveur WAPT n'est pas joignable lors de la mise à niveau, l'agent WAPT sera toujours capable d'installer les paquets mis en cache.

Les mises à jour de l'inventaire seront ensuite envoyées au serveur WAPT lorsque la connectivité réseau sera rétablie.

Les 5 objectifs de l'agent WAPT sont donc :

- Pour installer un paquet **base**, un **groupe** ou un **unité** s'il est disponible ;
- De supprimer les paquets obsolètes.
- Pour résoudre les dépendances et les conflits de paquets.
- Pour s'assurer que tous les paquets WAPT installés sont à jour par rapport à ceux stockés dans le dépôt.
- Pour mettre régulièrement à jour le serveur WAPT avec son état matériel et l'état des logiciels installés.

## 3.4 Comportement de l'agent WAPT

Un concept clé qui peut être difficile à comprendre est le comportement d'un agent WAPT lors de l'installation d'un paquet et les considérations qui l'entourent.

L'installation du paquet d'agents WAPT peut être divisée en étapes simples :

- Lors du déclenchement d'un **update**, l'agent télécharge les paquets *NEED-UPGRADE* ou *NEED-INSTALL* et les stocke dans le dossier cache.
- Lors du déclenchement d'un **upgrade**, l'agent décompresse les paquets dans un dossier temporaire.
- Le contenu du `setup.py` est analysé et stocké dans la base de données de l'agent WAPT située dans `C:\Program Files (x86)\wapt\db\waptdb.sqlite`.
- Le `setup.py` est exécuté et le logiciel est installé à partir des fichiers décompressés.
- En cas de succès : les paquets téléchargés et les fichiers dézippés sont supprimés. Un statut **OK** est renvoyé au serveur WAPT.
- En cas d'échec : les paquets téléchargés sont conservés et les fichiers dézippés sont supprimés. Un statut **ERROR** est renvoyé au serveur WAPT.

Ce comportement est important pour comprendre le cycle de vie d'un paquet installé.

Par exemple, lors du retrait d'un paquet, les étapes suivantes sont suivies :

- Le contenu du `setup.py` est extrait de la base de données de l'agent WAPT située dans `C:\Program Files (x86)\wapt\db\waptdb.sqlite`.
- L'agent WAPT recherche le `UninstallString` dans la base de données locale.
- Si elle est définie dans le `setup.py` copié dans la base de données locale lors de l'installation initiale du paquetage WAPT, la fonction `uninstall()` est exécutée.

Des étapes similaires sont reproduites lors de l'exécution de `session_setup` et `audit`.

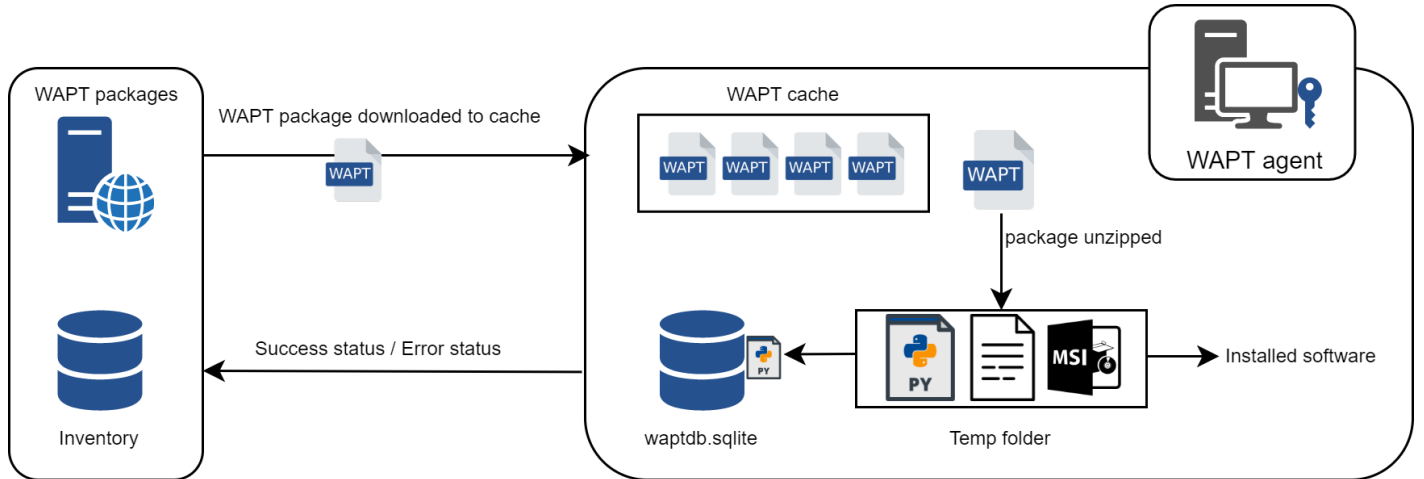


FIG. 5 – Comportement d’installation de WAPT

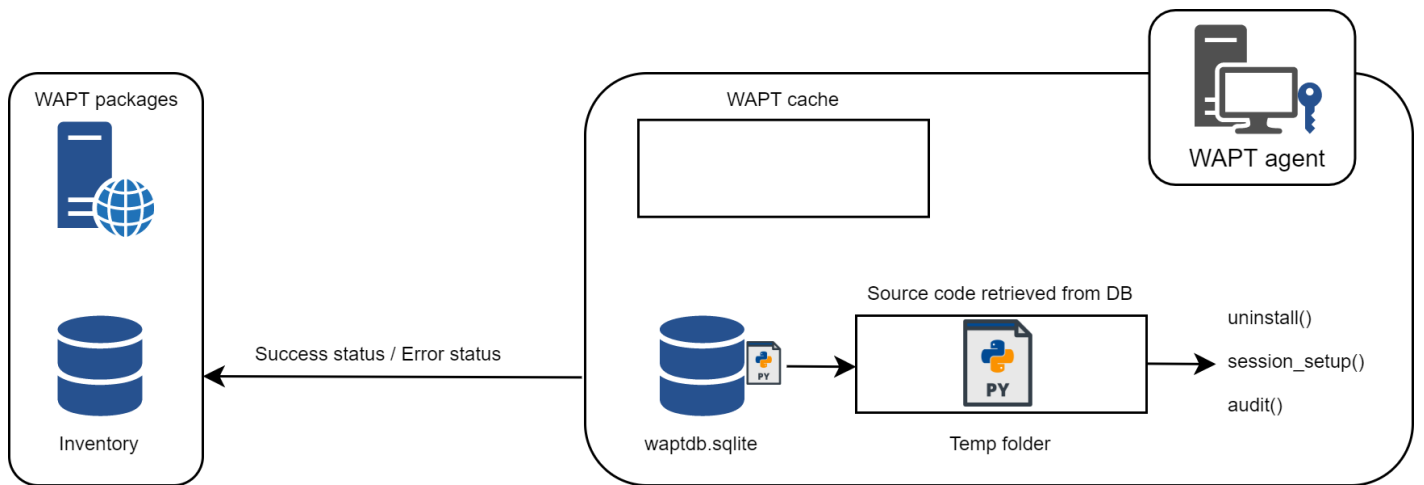


FIG. 6 – Comportement de l’agent WAPT avec la désinstallation, le session\_setup et l’audit

### 3.5 Diagramme complet du fonctionnement de WAPT

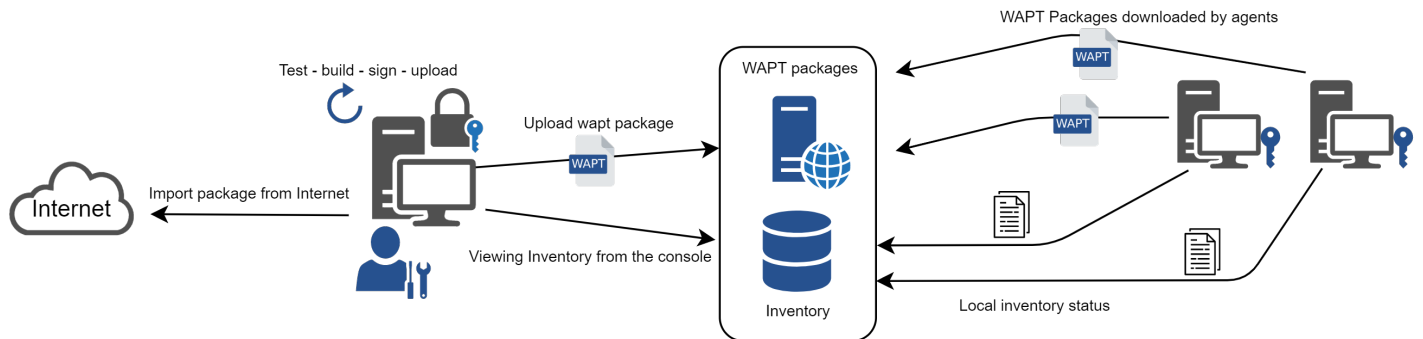


FIG. 7 – Mode de fonctionnement général du WAPT

Nous retrouvons ici le comportement commun de WAPT, depuis la duplication d'un paquet à partir d'un dépôt externe accessible sur Internet, jusqu'à son déploiement sur les machines du réseau.

Lire le diagramme dans le sens des aiguilles d'une montre :

- Importer des paquets depuis un dépôt externe (ou créer un nouveau paquet à partir de zéro).
- Tester, ensuite valider, puis construire et enfin signer le paquet.
- Télécharger le paquet sur le dépôt principal.
- Les paquets sont automatiquement téléchargés par les clients WAPT.
- Exécution des paquets selon la méthode sélectionnée :
  - L'*Administrateur* force l'**upgrade**.
  - L'*Administrateur* propose l'**upgrade** à l'*Utilisateur*.
  - Une tâche planifiée lance l'exécution de la mise à jour.
  - La mise à jour est exécutée à l'extinction de la machine.
  - L'*Utilisateur* choisit le bon moment pour lui-même (à l'arrêt ou en utilisant le *self-service*).
- Remontée des informations d'inventaire.
- Consultation de la remontée d'inventaire via la console.

## Architecture du serveur WAPT

L'architecture du serveur WAPT repose sur plusieurs rôles distincts :

- Le rôle de *dépôt* pour la distribution des packages.
- Le rôle *inventaire* et *serveur central* pour l'inventaire du matériel et des logiciels.
- Le rôle *proxy* pour relayer les actions entre la console WAPT et les agents WAPT.

### 4.1 Fonctionnement du dépôt WAPT

Tout d'abord, le serveur WAPT sert de dépôt de fichiers web.

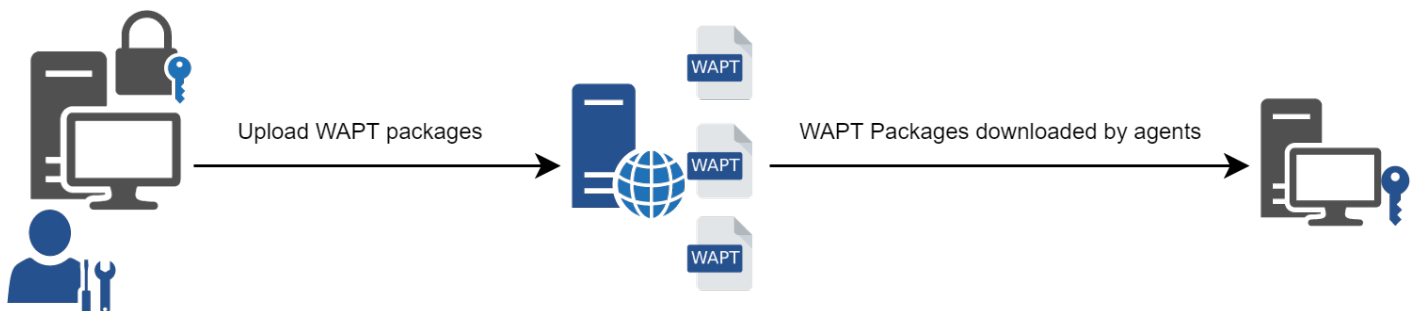


FIG. 1 – Fonctionnement du dépôt WAPT

- Ce rôle de dépôt est accompli par un serveur web Nginx.
- Le dépôt permet la distribution des paquets WAPT, des installateurs *waptagent* et *waptsetup*.
- Les paquets WAPT sont accessibles avec un navigateur web à l'adresse <https://srvwapt.mydomain.lan/wapt>.
- Les paquets **host** sont contenus dans un répertoire inaccessible par défaut (<https://srvwapt.mydomain.lan/wapt/wapt-host/>).

## 4.2 Rôle d'inventaire

Deuxièmement, le serveur WAPT sert de serveur d'inventaire.

Le serveur d'inventaire est un service passif qui collecte les informations que les agents WAPT lui envoient :

- Inventaire matériel.
- Inventaire logiciel.
- Statut des paquets WAPT.
- Etat des tâches (*running*, *pending*, *error*).

---

**Note :** Le service WAPT n'est pas actif dans le sens où il ne fait que recevoir des informations des clients. Par conséquent, si le serveur d'inventaire tombe en panne, l'inventaire se rétablira de lui-même à partir des rapports d'état d'inventaire reçus des agents WAPT déployés.

Dans la version **Discovery** de WAPT, l'accès aux données d'inventaire n'est possible que par la console WAPT.

WAPT **Enterprise** est livré avec des capacités *reporting*. En parallèle, il est possible de pousser l'inventaire WAPT vers l'outil ITSM *GLPI*.

---

## 4.3 Rôle de Proxy

Troisièmement, le serveur WAPT sert de proxy de commande.

Il sert de relais entre la console de gestion WAPT et les agents WAPT déployés.

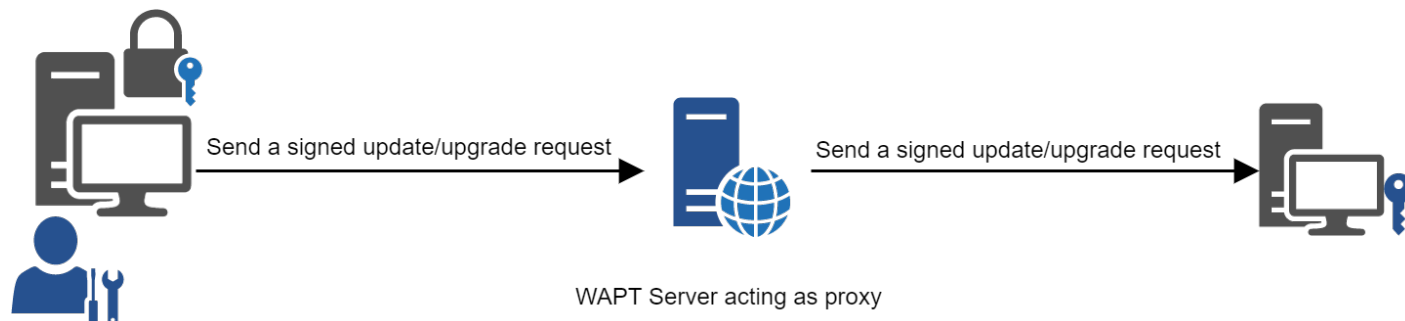


FIG. 2 – Fonctionnement du proxy WAPT

---

**Note :** Chaque action déclenchée sur un agent WAPT à partir du serveur est signée avec une clé privée. Sans une clé privée valide, il n'est pas possible de déclencher des actions à distance sur des appareils distants équipés de WAPT. Pour plus d'informations sur les actions à distance, veuillez vous référer à *signer des actions relayées aux agents WAPT*.

---

---

## Langage et environnement de développement WAPT

---

WAPT est construit en utilisant le langage `Python`.

**Attention :** Avec WAPT 2.0, le code interne de WAPT est passé à `python3`. Les paquets WAPT doivent MAINTENANT suivre la nouvelle syntaxe `python3`.

Consultez *cette documentation* pour vous aider à identifier les problèmes potentiels lors du passage de vos paquets existants de Python2 à Python3.

Tout environnement de développement rapide d'applications destiné au développement de Python convient.

Tranquil IT a développé quelques plugins spécifiques WAPT utiles pour l'IDE **PyScripter** (<https://sourceforge.net/projects/pyscripter>).

Tranquil IT recommande d'utiliser **PyScripter** pour développer des paquets WAPT pour Windows et **vscode** pour développer des paquets WAPT pour macOS et Linux.

### 5.1 La puissance de Python

Toute la puissance de **Python** peut être avantageusement mise à profit.

De nombreuses bibliothèques existent déjà en Python pour :

- Faire des boucles conditionnelles (si ... alors ... autrement ...).
- Copier, coller, déplacer des fichiers et des répertoires.
- Vérifier si les fichiers ou les répertoires existent.
- Vérifier si les clés de registre existent.
- Vérifier les droits d'accès, modifier les droits d'accès.
- La recherche d'informations sur des sources de données externes (LDAP, bases de données, fichiers, etc.).
- Et plus.

## 5.2 La puissance de WAPT

Les fonctions les plus couramment utilisées avec WAPT ont été simplifiées dans des bibliothèques appelées *Setuptools*.

Les fonctions **Setuptools** simplifient le processus de création et de test des paquets WAPT, validant ainsi les principaux objectifs de WAPT :

- **Ce qui était compliqué est rendu simple.**
- **Ce qui était simple est rendu trivial.**



---

## Comparaison des caractéristiques des versions de WAPT

---

### 6.1 Résumé des principes de fonctionnement de WAPT

- **WAPT est basé sur un agent qui n'autorise aucun port entrant ouvert** dans les pare-feu de la machine et qui initie un websocket bidirectionnel sécurisé avec le serveur pour permettre des rapports et des actions en temps réel.
- WAPT fonctionne avec des passerelles de données de confiance en utilisant une simple planification des tâches.
- WAPT fonctionne sur le principe de l'extraction progressive des mises à jour, puis de l'application des mises à niveau au moment opportun (fonctionne avec une bande passante faible ou intermittente, une latence élevée et des réseaux à forte instabilité).
- WAPT n'a pas besoin d'un Active Directory pour fonctionner (fonctionne aussi avec l'édition familiale de Windows); cependant, WAPT montrera la machine dans son emplacement Active Directory si l'hôte est joint à un AD.
- Méthodes pour déployer l'agent WAPT :
  1. En utilisant une GPO (Group Policy Object) ou un script Ansible.
  2. Après avoir téléchargé manuellement l'agent depuis le serveur WAPT ou en utilisant SSH (Secured Shell).
- Méthodes d'enregistrement des machines auprès du serveur WAPT :
  1. Automatiquement en utilisant le compte kerberos de la machine.
  2. Manuellement avec le login et le mot de passe WAPT *Superadmin*.
- Des mises à niveau peuvent être déclenchées :
  1. Lors de l'arrêt de la machine, c'est le mode standard.
  2. Par un Administrateur WAPT autorisé en cas d'urgence (ex : vulnérabilités critiques courant dans la nature).
  3. Par l'utilisateur au moment qu'elle choisit (ex : chariot de soins infirmiers 24/7 non utilisé pendant les pauses par un simple clic).
  4. Via une tâche planifiée s'exécutant à une heure prédéterminée (idéal pour les serveurs).
- La sécurité est assurée avec :
  1. La signature des paquets WAPT en utilisant la cryptographie asymétrique.
  2. L'authentification des hôtes par rapport au serveur WAPT en utilisant la cryptographie symétrique lors de l'enregistrement.

3. La confidentialité du serveur WAPT en utilisant les certificats des clients déployés par WAPT.
4. L'utilisation de ACL pour définir ce qu'un administrateur est autorisé à voir ou les actions qu'il est autorisé à effectuer en fonction de son certificat.

## 6.2 Liste des caractéristiques actuelles en date du 2024-01-09

**Attention :** Vous pouvez trouver sur Internet la mention d'une version GPLv3 **Community** de WAPT qui a été maintenue et supportée par Tranquil IT jusqu'à la version 1.8.2, soit jusqu'à environ juillet 2021.

La version **Community** du WAPT a été *forkée de manière amicale*. **Tranquil IT ne fournit plus aucun support, ni aucune maintenance, qu'elle soit gratuite ou payante sur WAPT =< 1.8.2**. Le support et la maintenance peuvent être obtenus auprès des opérateurs du *fork* à leurs tarifs et conditions.

**Tranquil IT est le seul auteur et le titulaire intégral des droits d'auteur de WAPT 1.8.2** et exigera des responsables de *friendly forks* qu'ils s'abstiennent d'utiliser le nom *WAPT* car la marque WAPT est déposée et protégée par l'Institut National de la Propriété Intellectuelle (INPI) en France et dans le monde.

TABLEAU 1 – Comparaison des caractéristiques entre les versions WAPT en date du 2024-01-09

Caractéristique	Enterprise	Discovery	Community
<b>Déploiement, mise à jour et suppression des logiciels</b> sur les hôtes	✓	✓	✓ <sup>1</sup>
Maintenance et support (voir note de bas de page pour les conditions)	Équipe Tranquil IT <sup>4</sup>	Forum Tranquil IT <sup>7</sup>	Communauté Open-Source
Sous licence	Propriétaire	Propriétaire	GPLv3
Limitation du nombre d'appareils	aucune limite	300	aucune limite
Version de Python utilisée dans le code et les paquets WAPT	3+ (actuel)	3+ (actuel)	2.7 (obsolète)
Déployer et mettre à jour les <b>configurations dans le contexte du SYSTÈME</b>	✓	✓	✓ <sup>2</sup>
Déployer et mettre à jour les <b>configurations dans le contexte de l'UTILISATEUR</b>	✓	✓	✓ <sup>2</sup>
Obtenez un <b>inventaire complet</b> du matériel, des logiciels et des paquets WAPT appliqués	✓	✓	✓
Bénéficier du <b>self-service différencié</b> (les utilisateurs autorisés peuvent installer les logiciels autorisés à partir du store de paquets WAPT autorisés)	✓	✗	✗
Bénéficiez de <b>Mises à jour Windows simplifiées</b> qui fonctionnent mieux qu'un WSUS standard (seules les KB requises sont téléchargées depuis Microsoft)	✓	✗	✗
Simplifiez et structurez votre charge de travail administrative en appliquant des paquets WAPT à vos UO (Unités d'Organisation)	✓	✗	✗
Configurer et gérer facilement les dépôts de secondaires WAPT <b>pour préserver la bande passante</b> pour les scénarios <i>Edge Computing</i>	✓	✗	✗
Accédez à des <b>paquets WAPT prêts à être déployés</b> pour des logiciels communs gratuits	✓	✓	✓ <sup>2</sup>
Travailler avec des <b>recettes python facilement vérifiables</b> pour l'installation, la mise à jour et la suppression de logiciels et de configurations	✓	✓	✓ <sup>2</sup>
Bénéficiez de <b>centaines d'assistants</b> pour simplifier le conditionnement des logiciels	✓ <sup>2</sup>	✓	✓ <sup>2</sup>
<b>Chiffrez vos données sensibles</b> pour le transport (clés de licence de logiciel, login, mot de passe, FQDN du serveur, informations API pour l'enregistrement du logiciel auprès du fournisseur, etc)	✓	✗	✗
Automatisez l'audit de vos configurations pour une conformité <b>facile, automatisée et toujours à jour</b>	✓	✗	✗
Profitez de la puissance de SQL intégrée à la console WAPT pour créer les <b>rapports dont vous avez besoin pour votre travail quotidien d'administrateur système ou dont votre organisation a besoin pour prendre des décisions budgétaires</b>	✓	✗	✗
Authentifiez vos <i>Administrateurs WAPT</i> avec <b>Active Directory, LDAP, ou avec leurs certificats personnels</b>	✓	✗	✗ <sup>3</sup>
Bénéficiez de rôles différenciés entre vos <i>Développeurs de Paquets</i> et vos <i>Déployeurs de Paquets</i> afin que vous puissiez <b>déléguer vos pouvoirs WAPT aux personnes les plus adéquates</b> (les développeurs de paquets connaissent les implications en matière de sécurité, les déployeurs connaissent les besoins des utilisateurs)	✓	✗	✗
Bénéficier du mode multi-tenant et multi-client avec les ACLs (Access Control Lists) pour les MSPs (Managed Service Providers) ou les grandes organisations multi-départementales ou internationales utilisant un mécanisme interne basé sur la PKI (Public Key Infrastructure) pour le périmètre autorisé	✓	✗	✗
Partage d'écran simple à utiliser pour l'assistance aux utilisateurs, conçu avec le même niveau de sécurité que les autres fonctionnalités WAPT (les utilisateurs peuvent demander).	✓	✗	✗
<b>Poursuite de la prise en charge de Windows XP</b> dans WAPT pour les machines-outils d'usine, les équipements médicaux des hôpitaux, les instruments de recherche coûteux et difficiles à remplacer, etc	✓ <sup>5</sup>	✗	✗
Mise à jour des paquets directement dans la console WAPT avec la fonction <code>update_package</code>	✓	✗	✗

## 6.3 Fonctionnalités à venir

Vous trouverez ci-dessous une liste de fonctionnalités que nous avons identifiées comme étant vraiment utiles à WAPT et à la communauté des utilisateurs de WAPT et sur lesquelles nous avons déjà commencé à travailler. Aucun calendrier n'est promis, restez à l'écoute, nous vous promettons seulement que nous travaillons très dur pour atteindre ces objectifs.

Caractéristique	En-ter-prise	Dis-co-very
Historique des actions effectuées via WAPT pour un rapport complet du cycle de vie de la maintenance d'un logiciel hôte	✓	✗
Authentification des administrateurs WAPT à l'aide de jetons cryptographiques (ex : cartes à puce)	✓	✗
Accès à des paquets WAPT prêts à être déployés ou des squelettes de paquets pour des logiciels d'entreprise sous licence (logiciels d'entreprise courants pour l'industrie, le secteur médical, les bureaux, les collectivités publiques, la cybersécurité, etc.)	✓	✗
Accès aux extensions de paquets WAPT prêtes à être déployées pour simplifier le blindage du bureau en utilisant Applocker ou équivalent	✓	✗
Vérifiez le paquet avec <a href="http://www.virustotal.com">www.virustotal.com</a>	✓	✓ <sup>7</sup>
Outil de déploiement d'images de systèmes d'exploitation intégré à WAPT	✓	✗

## 6.4 Principaux avantages fonctionnels de la version Entreprise de WAPT



WAPT **Discovery** est conçu pour vous permettre d'essayer gratuitement WAPT sur un périmètre limité et avec des fonctionnalités haut de gamme limitées.

Avec WAPT **Enterprise**, vous bénéficiez automatiquement des fonctions de base incluses dans WAPT pour vous aider à déployer, mettre à niveau et supprimer des logiciels et des configurations sur vos appareils Windows, Linux et MacOS, à partir d'une console centrale, avec de nombreux autres avantages.

WAPT est un modèle *libre*. La version **Enterprise** partage la même base de code que la version **Discovery**. Une clé de licence **Enterprise** active permet d'activer les fonctionnalités supplémentaires suivantes :

- **Authentification Active Directory**

ses développeurs de paquets WAPT, des déployeurs de paquets, des utilisateurs du self-service et pour l'enregistrement initial des agents WAPT auprès du serveur WAPT. En outre, l'affichage des appareils équipés de WAPT dans la console WAPT suit la même structure que la structure hiérarchique de l'Active Directory OU de l'organisation.

- **Séparation des rôles entre les développeurs de paquets et les déployeurs de paquets.**

De cette façon, les équipes informatiques centrales peuvent construire les progiciels parce qu'elles connaissent les directives de sécurité de l'Organisation, et les équipes informatiques locales peuvent déployer les progiciels WAPT parce qu'elles connaissent les besoins de leur base d'utilisateurs.

1. WAPT =< 1.8.2 implémente python2.7, il n'y a donc aucune garantie que les paquets conçus pour python3 fonctionneront.
4. Un volume minimal de licences doit être souscrit afin de bénéficier du support téléphonique de Tranquil IT pour l'exploitation quotidienne du logiciel. Une assistance supplémentaire payante est disponible pour vous aider à répondre à vos besoins en matière de packaging WAPT. Le support du forum est fourni sans garantie ni délai et peut être assuré par des utilisateurs **Enterprise** ou **Discovery** non affiliés à Tranquil IT.
2. La version Enterprise intègre plus de fonctions SetupHelper que les versions **Community** et **Discovery**.
3. Dans les versions **Community** et **Discovery**, le mot de passe WAPT *SuperAdmin* est partagé entre les personnes qui gèrent le serveur WAPT.
5. Windows XP ne fonctionne pas avec Python > 2.7. Une branche spéciale de WAPT sera donc gelée avec la dernière version de l'agent WAPT fonctionnant avec 2.7. Cette version de l'agent sera bien sûr exclue de la cible d'évaluation lors des futures certifications de sécurité.
7. Seulement pour les paquets sur le store WAPT certifié par Tranquil IT. Pour bénéficier de virustotal pour vos propres paquets, la version Entreprise est nécessaire.

Une telle séparation est mise en œuvre à l'aide de jeux de clés différenciés (c'est-à-dire des certificats SSL **Code Signing** pour les développeurs de paquets et des certificats SSL **Simple** pour les *déploieurs* de paquets) et avec des rights ACL.

— **ACL.**

Les ACLs sont gérées par le *SuperAdmin* pour autoriser ou restreindre les *Administrators* WAPT à visualiser des informations ou à effectuer des actions uniquement sur un sous-ensemble de dispositifs enregistrés auprès du serveur WAPT.

Les processus d'identification et d'authentification reposent soit sur l'utilisation d'Active Directory, de LDAP ou de certificats. Les autorisations accordées aux administrateurs sont gérées dans la base de données du serveur WAPT. Le périmètre des dispositifs sur lesquels les droits sont accordés est défini par le certificat de l'administrateur déployé.

Cette fonction est particulièrement utile pour les grandes organisations multinationales, les administrations centrales avec de grands bureaux régionaux ou pour les MSP (Managed Service Providers) qui souhaitent centraliser la gestion de plusieurs clients tout en permettant à leurs clients finaux d'effectuer certaines tâches de gestion quotidiennes.

— **Libre service différencié.**

WAPT Enterprise vous permet d'appliquer des listes de paquets autorisés à des groupes d'utilisateurs dans Active Directory. Les utilisateurs autorisés sont libres d'installer des paquets qualifiés à partir de leur liste de paquets approuvés sans avoir à soumettre un ticket à leurs équipes informatiques.

Cette fonction est conçue pour offrir aux *Utilisateurs* le sentiment de liberté et d'autonomie qu'ils craignent de perdre dans les environnements gérés, tout en permettant aux RSSI d'appliquer des règles de sécurité strictes à l'aide d'une méthode telle que SRP (Software Restriction Policies), également connue sous le nom de *Applocker*.

— **WAPT WUA.**

WAPT permet de gérer les mises à jour de Windows sur vos terminaux Windows.

Le WAPT WUA est conçu pour fonctionner immédiatement, ménager votre stockage et préserver votre bande passante pour vos besoins de production.

— **Rapports avancés pour les équipes de l'entreprise.**

Ces rapports complètent les rapports opérationnels déjà disponibles dans la console WAPT ; les rapports aident les opérateurs WAPT à démontrer leur efficacité avec WAPT pour assurer un plus grand niveau de sécurité et de conformité pour leurs réseaux, systèmes, logiciels et applications.

— **Configuration dynamique du dépôt.**

À partir de WAPT 1.8, la réplication de référentiel peut être activée en utilisant un agent WAPT installé sur une machine existante, une appliance dédiée ou une machine virtuelle.

Le rôle de réplication est déployé par le biais d'un paquet WAPT qui active le serveur web **Nginx** et configure la planification, les types de paquets, la synchronisation des paquets, et bien plus encore.

Cette fonction permet aux agents WAPT de trouver dynamiquement le dépôt WAPT disponible le plus proche à partir d'une liste de règles stockées sur le serveur WAPT.

— **Intégration avec GLPI**

GLPI est une solution populaire ITSM pour la gestion des tickets, des incidents et des actifs.

WAPT peut maintenant envoyer de manière optionnelle un ensemble minimum d'informations utiles à un serveur GLPI.

## 6.5 Cas d'utilisation ciblés de WAPT Enterprise

La version Entreprise de WAPT est particulièrement recommandée pour les organisations :

- Qui gèrent de grandes bases installées de dispositifs (généralement plus de 300 unités).
- Qui sont répartis géographiquement avec de nombreuses filiales ou sites de production.
- Qui exigent une forte traçabilité des actions effectuées sur la base installée de dispositifs pour des raisons d'audit ou de sécurité.
- Qui accordent de l'importance à des solutions sécurisées et éprouvées dans leur recherches IT.

## 6.6 Description des services disponibles avec un contrat WAPT Enterprise

### 6.6.1 Accès aux futures améliorations de WAPT Enterprise

En souscrivant à un contrat WAPT **Enterprise** et en maintenant votre abonnement valide, vous bénéficiez des améliorations futures apportées au cœur de WAPT et vous bénéficiez automatiquement de toutes les améliorations futures de la version WAPT **Enterprise**.

L'expiration de votre abonnement fera automatiquement basculer votre instance WAPT vers sa version **Discovery** correspondante. Les fonctions avancées disponibles uniquement dans la version **Enterprise** ne seront plus accessibles et aucune action autre que la suppression d'hôtes à partir de la console ne sera autorisée tant que le nombre d'hôtes ne sera pas passé en dessous de 300.

### 6.6.2 Assistance téléphonique directe pour votre utilisation quotidienne de WAPT

Lorsque votre abonnement **dépasse un certain volume**, Tranquil IT, le créateur de WAPT, vous offre un accès privilégié à son équipe d'experts et de développeurs WAPT.

Nous vous donnons accès à une hot-line téléphonique dédiée avec une réponse directe pour satisfaire vos besoins d'assistance en **anglais** et **français**.

Nous nous engageons à vous fournir rapidement des réponses fiables et pertinentes sur le périmètre souscrit.

En souscrivant ou en renouvelant votre contrat WAPT **Enterprise**, vous recevrez une notification indiquant les modalités pratiques d'accès à notre support.

**Attention :** Le support ne concerne que l'utilisation dans votre Organisation du logiciel WAPT **Enterprise**, un support supplémentaire pour l'adaptation, la personnalisation, le débogage ou la création de paquets personnalisés WAPT peut être obtenu avec des tickets de support prépayés.

Jusqu'à trois personnes de votre *Organisation* peuvent communiquer avec notre support direct.

---

**Note :** Pour plus d'informations, contactez l'équipe commerciale de Tranquil IT.

---

### 6.6.3 Prix et accès préférentiel à la formation WAPT

Vous pouvez choisir de former votre équipe informatique sur n'importe quelle particularité de WAPT.

Les abonnés WAPT **Enterprise** bénéficient d'un accès privilégié aux conseillers en formation de Tranquil IT et d'une réduction de 50% sur les prix des formations standard.

---

**Note :** Pour plus d'informations, contactez l'équipe commerciale de Tranquil IT.

---

---

## Prérequis d'installation

---

Vous devez prendre en considération quelques points de sécurité afin de tirer tous les avantages possibles du WAPT :

- Si vous êtes familier avec Linux, nous vous conseillons d'installer le serveur WAPT directement sur CentOS en suivant les recommandations de sécurité de l'ANSSI ou les [recommandations de l'agence de cyberdéfense de votre état](#).
- Bien que le serveur WAPT ne soit pas conçu pour être un actif sensible, nous recommandons qu'il soit installé sur une **machine dédiée** (physique ou virtuelle).

**Attention :** Dans toutes les étapes de la documentation, vous n'utiliserez aucun accent ni caractères spéciaux pour :

- le login des utilisateurs ;
- le chemin de la clé privée et du certificat ;
- le CN (Common Name) ;
- le chemin d'installation de WAPT ;
- les noms de groupe ;
- le nom des hôtes ou le nom du serveur ;
- le chemin vers le répertoire C:\waptdev.

### 7.1 Préconisations matérielles

Le serveur WAPT peut être installé soit sur un serveur virtuel, soit sur un serveur physique.

Les recommandations de RAM et de CPU sont :

Taille de parc	CPU	RAM	Optimisation serveur à appliquer
De 0 a 300 postes	2 CPU	2024 Mio	Non
De 300 a 1000 postes	4 CPU	4096 Mio	Oui
De 1000 a 3000 postes	8 CPU	8192 Mio	Oui
A partir de 3000 postes et plus	16 CPU	16384 Mio	Oui

- Un minimum de 10 Go d'espace libre est nécessaire pour le système, la base de données et les fichiers de journalisation. **Pour de meilleures performances, Tranquil IT recommande que la base de données soit stockée sur des supports rapides, tels que des disques SSD ou des SSD sur PCIe.**
- L'exigence globale en matière de disque dépendra du nombre et de la taille de vos paquets WAPT (logiciels) que vous stockerez sur votre dépôt principal, 30 Go étant un bon début. Il n'est pas strictement nécessaire de stocker les paquets WAPT sur des disques rapides.
- Enfin, nous avons connaissance d'utilisateurs disposant de serveurs équipés de multiples interfaces réseau 10Gbps déployant à pleine vitesse des paquets de mise à jour massifs de Katia, National Instruments et Solidworks sur leur LAN (Local Area Network).

## 7.2 Préconisations logicielles

### 7.2.1 Système d'exploitation

Le serveur WAPT est disponible sur Linux et Windows :

- Pour Linux, **Debian 11, Red Hat 7 / 8 et dérivés, Ubuntu server LTS 20.04** la version 64 bit est supporté. Il n'est pas obligatoire d'utiliser une distribution Linux serveur, mais utilisez une distribution **non graphique**.

---

**Note :** SELINUX est supporté mais pas obligatoire.

---

- Pour Windows, le serveur WAPT peut être installé sur **Windows Server** version 64 bits supportée par Microsoft (Win2012r2, Win2k16 ou Win2k19). Selon votre besoin, il peut également être installé sur une version récente de Win10 Pro/Ent (20H2 ou plus).

Le serveur WAPT ne fonctionnera que sur un système basé sur **64bit**.

### 7.2.2 Ouverture de ports

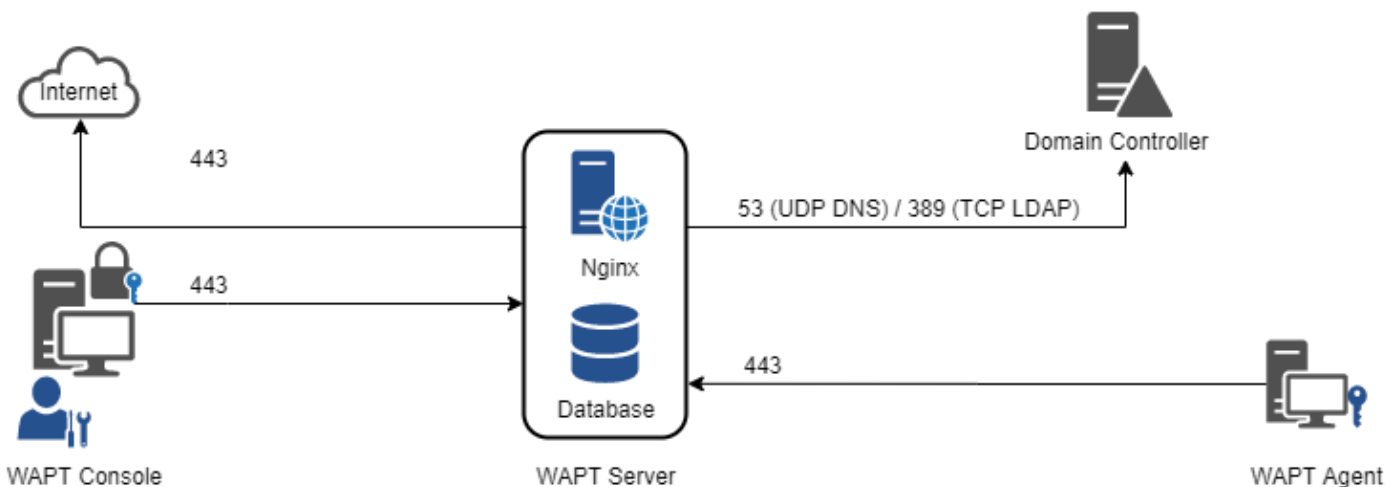


Fig. 1 – Diagramme des flux de données de WAPT

Comme vous pouvez le constater, seuls les ports **80** et **443** doivent être ouverts pour les connexions entrantes car les frameworks WAPT fonctionnent avec des websockets initiés par les agents WAPT.



**Entrant**

Proto- cole	Numéro de port	Source	Destination	Description
<i>TCP</i>	<b>80</b>	Tous les agents WAPT	Serveur WAPT	Connexion Websocket (non-sécurisé) pour télécharger les paquets et les KB
<i>TCP</i>	<b>443</b>	Tous les agents WAPT	Serveur WAPT	Connexion Websocket pour télécharger les paquets et les KB

**Sortant**

Proto- cole	Numéro de port	Source	Destination	Description
<i>TCP</i>	<b>80</b>	Serveur WAPT	Internet	Connexion Websocket (non-sécurisé) pour télécharger les paquets, wsusscn2.cab et les KB
<i>TCP</i>	<b>443</b>	Serveur WAPT	Tous les agents WAPT	Connexion Websocket pour télécharger les paquets, wsusscn2.cab et les KB
<i>TCP</i>	<b>80</b>	Serveur WAPT	Dépôt Linux (pour les serveurs Linux)	Mise à jour pour les paquets
<i>TCP</i>	<b>443</b>	Serveur WAPT	Dépôt Linux (pour les serveurs Linux)	Mise à jour pour les paquets
<i>TCP</i>	<b>53</b>	Serveur WAPT	Contrôleur de domaine	Résolution DNS
<i>TCP</i>	<b>389</b>	Serveur WAPT	Contrôleur de domaine	Authentification LDAP
<i>TCP</i>	<b>636</b>	Serveur WAPT	Contrôleur de domaine	Authentification LDAP
<i>UDP</i>	<b>123</b>	Serveur WAPT	Contrôleur de domaine	NTP



### 8.1 Configurer les DNS de l'Organisation pour WAPT

---

**Note :** La configuration DNS n'est pas obligatoire, elle est fortement recommandée.

---

Afin de faciliter la gestion de votre installation WAPT, il est fortement recommandé de configurer le serveur *DNS* pour inclure le champ *A* ou le champ *CNAME* comme ci-dessous :

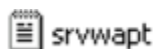
- *srvwapt.mydomain.lan.*
- *wapt.mydomain.lan.*

Remplacer *mydomain.lan* par le suffixe *DNS* utilisé sur votre réseau.

Ces champs seront utilisés par les agents WAPT pour trouver le serveur WAPT ou un dépôt secondaire WAPT de proximité sur le réseau.

### 8.2 Configurer les champs DNS avec les « Outils d'administration de serveur distant » Microsoft (RSAT).

- Le champ *A* pointe vers l'adresse IP du serveur WAPT.



srvwapt

Hôte (A)

192.168.149.37

Vous pouvez maintenant installer votre serveur WAPT sur l'OS de votre choix :

- *Installer le serveur WAPT sur GNU / Linux Debian.*
- *Installer le serveur WAPT sur CentOS / RedHat.*
- *Installer le serveur WAPT sur Windows.*



---

## Installer le serveur WAPT sur Linux Debian

---

---

**Indication :** Cette partie est seulement pour Debian, pour Ubuntu vous reporter à *ce point*.

---

### 9.1 Configuration du serveur GNU/Linux Debian

Afin d'installer un (physique ou virtuel), veuillez vous référer au [Guide d'installation de Debian GNU/Linux](#).

**Avertissement :**

- Installer la version 64bits.
- Installez le serveur sans interface graphique.

**Danger :** Pour le Serveur WAPT **Nginx** est le **SEUL** server web supporté. **Apache n'est plus supporté par WAPT.**

#### 9.1.1 Configurer les paramètres réseau

Les différents paramétrages préconisés ci-dessous ne sont pas spécifiques à WAPT; vous devrez les adapter à votre environnement.

Modifiez les fichiers suivants afin d'obtenir une stratégie de nommage (*FQDN*) et d'adressage réseau appropriée.

Dans l'exemple suivant :

- le *FQDN* est *srvwapt.mydomain.lan*;
- le nom court du serveur WAPT est *srvwapt*;
- le suffixe *DNS* est *mydomain.lan*;
- l'adresse IP est *10.0.0.10/24*;

## 9.1.2 Configurer le nom du serveur WAPT

**Indication :** Le nom du serveur WAPT ne doit pas dépasser **15 caractères** (limite liée au sAMAccountName dans Active Directory).

---

Le nom du serveur doit être un nom FQDN (Fully Qualified Domain Name), c'est à dire à la fois le nom de machine et le suffixe DNS.

- Modifier le fichier `/etc/hostname` et y renseigner le nom *FQDN* du serveur.

```
# /etc/hostname of the WAPT server
srvwapt.mydomain.lan
```

- Configurez le fichier `/etc/hosts`, assurez-vous de mettre à la fois le *FQDN* et le nom court du serveur.

```
# /etc/hosts of the WAPT server
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.0.10   srvwapt.mydomain.lan    srvwapt
```

**Indication :**

- Sur la ligne définissant l'adresse IP du serveur DNS, veillez à avoir l'IP du serveur (pas 127.0.0.1), puis le *FQDN*, puis le nom court.
  - Ne changez pas la ligne avec *localhost*.
- 

## 9.1.3 Configurer l'adresse IP du serveur WAPT

- Configurez l'adresse IP du serveur WAPT dans `/etc/network/interfaces`.

```
# /etc/network/interfaces of the WAPT server
auto eth0
iface eth0 inet static
    address 10.0.0.10
    netmask 255.255.255.0
    gateway 10.0.0.254
```

- Appliquez la configuration réseau en redémarrant la machine avec un **reboot**.

```
reboot
```

- Si cela n'a pas déjà été fait, créez l'entrée *DNS* pour le serveur WAPT dans l'Active Directory de l'*Organisation*.
- Après le redémarrage, configurez la langue du système en anglais afin d'avoir des journaux non localisés pour faciliter la recherche des erreurs courantes.

```
apt install locales-all -y
localectl set-locale LANG=en_US.UTF-8
localectl status
```

- Vérifiez si le service *ntp* est installé, démarré et si l'heure est correcte.

```
dpkg -l | grep ntp
service ntp status
date
```

**Indication :** Si le paquet NTP n'est pas installé.

```
apt install ntp -y
systemctl enable ntp
systemctl start ntp
```

— Mettre à jour Debian.

```
apt update && apt upgrade
```

— Installer `systemd`.

```
apt install systemd -y
```

— Installer les autorités de certifications livrées avec les navigateurs Mozilla.

```
apt install ca-certificates -y
```

— Redémarrer le serveur.

```
reboot
```

Le serveur Debian est maintenant prêt. Vous pouvez maintenant passer à l'étape suivante et *installer le serveur WAPT sur votre Debian*.

**Attention :** La procédure de mise à jour est différente de l'installation. Pour une mise à jour, rendez-vous sur *la documentation pour mettre à jour le serveur WAPT*.

L'installation de la partie serveur de WAPT se décompose en plusieurs étapes :

- Configurer les dépôts.
- Installer les paquets Linux complémentaires.
- Installation et provisionnement de la base de données PostgreSQL.
- Post-configuration du serveur WAPT.

**Note :** Les paquets du serveur WAPT et le dépôt sont signés par Tranquil IT et il est nécessaire d'obtenir la clé publique gpg ci-dessous afin d'éviter les messages d'avertissement pendant l'installation.

## 9.2 Configuration du dépôt DEB

La configuration des dépôts en **WAPT Enterprise** et en **WAPT Discovery** diffère. **Assurez-vous de choisir la bonne procédure!!**

### 9.2.1 Discovery

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition **WAPT Discovery**. Pour **WAPT Enterprise** Edition, veuillez vous référer au bloc suivant.

---

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/fr/doc-1.8/>

---

### 9.2.2 Enterprise

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition **WAPT Enterprise**. Pour **WAPT Discovery** Edition, veuillez vous référer au bloc précédent.

---

— Installez `apt-transport-https` pour l'utilisation du https.

```
apt install apt-transport-https lsb-release gnupg -y
```

— Récupération de la clé `.gpg` et ajout du dépôt Tranquil'iT.

```
wget -O - https://wapt.tranquil.it/debian/tiswapt-pub.gpg | apt-key add -  
echo "deb https://srvwapt-pro.tranquil.it/entreprise/debian/wapt-2.1/ $(lsb_release -c -s) main" > /  
→etc/apt/sources.list.d/wapt.list
```

— Créez `wapt.conf` dans `/etc/apt/auth.conf.d` pour stocker vos informations de connexion.

---

**Indication :** Remplacez `user` et `password` pour accéder au dépôt WAPT Enterprise, par ceux fournis par notre service commercial.

---

```
cat > /etc/apt/auth.conf.d/wapt.conf <<EOF  
machine srvwapt-pro.tranquil.it  
login user  
password password  
EOF
```

— Appliquez les bonnes ACLS sur `wapt.conf`.

```
chmod 600 /etc/apt/auth.conf.d/wapt.conf
```



## 9.3 Installer les paquets du serveur WAPT

```
export DEBIAN_FRONTEND=noninteractive
apt update
apt install tis-waptserver tis-waptsetup -y
unset DEBIAN_FRONTEND
```

## 9.4 Post-configuration

**Attention :** Pour que le post-configuration fonctionne correctement, vous devez d'abord avoir un *hostname* pour le serveur WAPT. Pour vérifier, utilisez la commande `echo $(hostname)` qui doit retourner l'adresse DNS qui sera utilisé par les agents WAPT sur les ordinateurs clients.

**Avertissement :** Le script de post-configuration réécrit la configuration de nginx.

Si vous utilisez une *configuration spéciale*, sauvegardez votre fichier `wapt.conf` avec la commande

```
cp /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-available/wapt.conf.old
```

Il sera nécessaire de réécrire la configuration après la post-configuration avec la commande :

```
cp /etc/nginx/sites-available/wapt.conf.old /etc/nginx/sites-available/wapt.conf
```

**Indication :** Ce script de post-configuration doit être exécuté en tant que **root**.

— Lancez le script.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

— Cliquez sur *Oui* pour lancer le script de post-configuration.

```
do you want to launch post configuration tool?
```

```
< yes >      < no >
```

— Choisissez un mot de passe (si ce n'est pas déjà défini) pour le compte *SuperAdmin* du serveur WAPT (longueur minimale de 10 caractères).

```
Please enter the wapt server password (min. 10 characters)
```

```
*****
```

```
< OK >      < Cancel >
```

— Confirmer le mot de passe.

Please enter the server password again:

\*\*\*\*\*

< OK >            < Cancel >

- Choisissez le mode d'authentification pour l'enregistrement initial des agents WAPT :
  - Le choix #1 permet d'enregistrer les ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistré.
  - Le choix #2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer plus tard).
  - Le choix #3 n'active pas le mécanisme d'authentification kerberos pour l'enregistrement initial des machines équipées de WAPT. Le serveur WAPT va demander un identifiant et mot de passe pour chaque machine qui s'enregistre.

WaptAgent Authentication type?

- 
- 1 Allow unauthenticated registration
  - 2 Enable kerberos authentication required **for** machines registration.  
Registration will ask **for** password **if** kerberos not available
  - 3 Disable kerberos but registration require strong authentication
- 

< OK >            < Cancel >

- Sélectionnez *OK* pour démarrer le serveur WAPT.

Press OK to start waptserver

< OK >

- Sélectionnez *Oui* pour configurer Nginx.

Do you want to configure nginx?

< Yes >            < No >

- Remplissez le *FQDN* du serveur WAPT.

FQDN **for** the WAPT server (eg. wapt.example.com)

-----  
wapt.mydomain.lan  
-----

< OK >            < Cancel >

- Sélectionnez *OK* et un certificat auto-signé sera généré, cette étape peut prendre un certain temps.

Generating DH parameters, 2048 bit long safe prime, generator 2 This is going to take a long time  
.....+.....+.....

Nginx est maintenant configuré, sélectionnez *OK* pour redémarrer **Nginx** :

The Nginx config is **done**.  
We need to restart Nginx?

< OK >

Le post-confoguration est maintenant terminé.

Postconfiguration completed.  
Please connect to <https://wapt.mydomain.lan/> to access the server.

< OK >

Liste des options du script de post-configuration :

Options	Description
<code>--force-https</code>	Configurer <b>Nginx</b> de sorte que <i>le port 80 soit redirigé de façon permanente vers le 443</i>

Le serveur est maintenant prêt. Vous pouvez consulter la documentation sur *l'installation de la console WAPT*.



---

## Installation du serveur WAPT sur Ubuntu

---

### 10.1 Configuration du serveur GNU/Linux Ubuntu

Afin d'installer un (physique ou virtuel), veuillez vous référer au [Guide d'Installation d'Ubuntu GNU/Linux](#).

**Avertissement :**

- Installez la version **64bit**.
- Installez un serveur sans interface graphique.
- Seulement la version LTS est supporté par WAPT.

**Danger : Nginx** est le **seul** serveur web supporté pour WAPT. **Apache sur Linux n'est plus supporté par WAPT.**

#### 10.1.1 Configuration des paramètres réseau

Les différents paramètres présentés ci-dessous ne sont pas spécifiques à WAPT, vous pouvez les adapter en fonction de votre environnement.

Modifiez les fichiers suivants afin d'obtenir une stratégie de nommage (*FQDN*) et d'adressage réseau appropriée.

Dans l'exemple suivant :

- le nom *FQDN* est *srvwapt.mydomain.lan* ;
- le nom court du serveur WAPT est *srvwapt* ;
- le suffixe *DNS* est *mydomain.lan* ;
- l'adresse IP est *10.0.0.10/24* ;

## 10.1.2 Configurez le nom du serveur WAPT

**Indication :** Le nom court du serveur WAPT ne doit pas dépasser **15 caractères** (la limite est due à la restriction *sAMAccountName* dans Active Directory).

---

Le nom du serveur WAPT doit être un FQDN, c'est-à-dire qu'il comporte à la fois le nom du serveur et le suffixe DNS.

- Modifiez le fichier `/etc/hostname` et écrivez le *FQDN* du serveur.

```
# /etc/hostname of the WAPT server
srvwapt.mydomain.lan
```

- Configurez le fichier `/etc/hosts`, assurez-vous de mettre à la fois le *FQDN* et le nom court du serveur.

```
# /etc/hosts of the WAPT server
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.0.10   srvwapt.mydomain.lan    srvwapt
```

**Indication :**

- Sur la ligne définissant l'adresse IP du serveur DNS, veillez à avoir l'IP du serveur (pas 127.0.0.1), puis le *FQDN*, puis le nom court.
  - Ne changer pas la ligne avec *localhost*.
- 

## 10.1.3 Configuration de l'adresse IP du serveur WAPT

- Configurez l'adresse IP du serveur WAPT dans le `/etc/network/interfaces`.

```
# /etc/network/interfaces of the WAPT server
auto eth0
iface eth0 inet static
    address 10.0.0.10
    netmask 255.255.255.0
    gateway 10.0.0.254
```

- Appliquez la configuration réseau en redémarrant la machine avec un **reboot**.

```
reboot
```

- Si cela n'a pas déjà été fait, créez l'entrée *DNS* pour le serveur WAPT dans l'Active Directory de l'*Organisation*.
- Après le redémarrage, configurez la langue du système en anglais afin d'avoir des journaux non localisés pour faciliter la recherche des erreurs courantes.

```
apt install locales-all -y
localectl set-locale LANG=en_US.UTF-8
localectl status
```

- Vérifiez si le service *ntp* est installé, démarré et si l'heure est correcte.
-

```
dpkg -l | grep ntp
service ntp status
date
```

**Indication :** Si le paquet NTP n'est pas installé.

```
apt install ntp -y
systemctl enable ntp
systemctl start ntp
```

— Mise à jour d'Ubuntu.

```
apt update && apt upgrade
```

— Installer `systemd`.

```
apt install systemd -y
```

— Installer les autorités de certification livrées avec le navigateur Mozilla.

```
apt install ca-certificates -y
```

— Redémarrer le serveur.

```
reboot
```

Le serveur Ubuntu est maintenant prêt. Vous pouvez maintenant passer à l'étape suivante et *installer WAPT sur votre Ubuntu*.

**Attention :** La procédure de mise à niveau est différente de l'installation. Pour la mise à niveau, veuillez vous reporter à *la documentation sur la mise à niveau du serveur WAPT*.

L'installation du serveur WAPT nécessite quelques étapes :

- Configuration des dépôts.
- Installation des paquets Linux supplémentaires.
- Installation et provisionnement de la base de données PostgreSQL.
- Post-configuration du serveur WAPT.

**Note :** Les paquets du serveur WAPT et le dépôt sont signés par Tranquil IT et il est nécessaire d'obtenir la clé publique gpg ci-dessous afin d'éviter les messages d'avertissement pendant l'installation.

## 10.2 Configurer les dépôts DEB

La configuration des dépôts en **WAPT Enterprise** et en **WAPT Discovery** diffère. **Assurez-vous de choisir la bonne procédure!**

### 10.2.1 Discovery

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition **WAPT Discovery**. Pour **WAPT Enterprise** Edition, veuillez vous référer au bloc suivant.

---

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>

---

### 10.2.2 Enterprise

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition **WAPT Enterprise**. Pour **WAPT Discovery** Edition, veuillez vous référer au bloc précédent.

---

— Installez `apt-transport-https` pour l'utilisation du https.

```
apt install apt-transport-https lsb-release gnupg -y
```

— Récupération de la clé `.gpg` et ajout du dépôt Tranquil'iT.

```
wget -O - https://wapt.tranquil.it/debian/tiswapt-pub.gpg | apt-key add -  
echo "deb https://srvwapt-pro.tranquil.it/entreprise/ubuntu/wapt-2.1/ $(lsb_release -c -s) main" > /  
→etc/apt/sources.list.d/wapt.list
```

— Créez `wapt.conf` dans `/etc/apt/auth.conf.d` pour stocker vos informations de connexion.

---

**Indication :** Remplacez `user` et `password` pour accéder au dépôt WAPT Enterprise, par ceux fournis par notre service commercial.

```
cat > /etc/apt/auth.conf.d/wapt.conf <<EOF  
machine srvwapt-pro.tranquil.it  
login user  
password password  
EOF
```

— Appliquez les ACL correctes sur `wapt.conf`.

```
chmod 600 /etc/apt/auth.conf.d/wapt.conf
```



## 10.3 Installation les paquets du serveur WAPT

```
export DEBIAN_FRONTEND=noninteractive
apt update
apt install tis-waptserver tis-waptsetup -y
unset DEBIAN_FRONTEND
```

## 10.4 Post-configuration

**Attention :** Pour que le post-configuration fonctionne correctement, vous devez d'abord avoir un *hostname* pour le serveur WAPT. Pour vérifier, utilisez la commande `echo $(hostname)` qui doit retourner l'adresse DNS qui sera utilisé par les agents WAPT sur les ordinateurs clients.

**Avertissement :** Le script de post-configuration réécrit la configuration de nginx.

Si vous utilisez une *configuration spéciale*, sauvegardez votre fichier `wapt.conf` avec la commande

```
cp /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-available/wapt.conf.old
```

Il sera nécessaire de réécrire la configuration après la post-configuration avec la commande :

```
cp /etc/nginx/sites-available/wapt.conf.old /etc/nginx/sites-available/wapt.conf
```

**Indication :** Ce script de post-configuration doit être exécuté en tant que **root**.

— Lancez le script.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

— Cliquez sur *Oui* pour lancer le script de post-configuration.

```
do you want to launch post configuration tool?
```

```
< yes >          < no >
```

— Choisissez un mot de passe (si ce n'est pas déjà défini) pour le compte *SuperAdmin* du serveur WAPT (longueur minimale de 10 caractères).

```
Please enter the wapt server password (min. 10 characters)
```

```
*****
```

```
< OK >          < Cancel >
```

— Confirmer le mot de passe.

Please enter the server password again:

\*\*\*\*\*

< OK >            < Cancel >

- Choisissez le mode d'authentification pour l'enregistrement initial des agents WAPT :
  - Le choix #1 permet d'enregistrer les ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistré.
  - Le choix #2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer plus tard).
  - Le choix #3 n'active pas le mécanisme d'authentification kerberos pour l'enregistrement initial des machines équipées de WAPT. Le serveur WAPT va demander un identifiant et mot de passe pour chaque machine qui s'enregistre.

WaptAgent Authentication type?

- 
- 1 Allow unauthenticated registration
  - 2 Enable kerberos authentication required **for** machines registration.  
Registration will ask **for** password **if** kerberos not available
  - 3 Disable kerberos but registration require strong authentication
- 

< OK >            < Cancel >

- Sélectionnez *OK* pour démarrer le serveur WAPT.

Press OK to start waptserver

< OK >

- Sélectionnez *Oui* pour configurer Nginx.

Do you want to configure nginx?

< Yes >            < No >

- Remplissez le *FQDN* du serveur WAPT.

FQDN **for** the WAPT server (eg. wapt.example.com)

-----  
wapt.mydomain.lan  
-----

< OK >            < Cancel >

- Sélectionnez *OK* et un certificat auto-signé sera généré, cette étape peut prendre un certain temps.

Generating DH parameters, 2048 bit long safe prime, generator 2 This is going to take a long time  
.....+.....+.....

Nginx est maintenant configuré, sélectionnez *OK* pour redémarrer **Nginx** :

The Nginx config is **done**.  
We need to restart Nginx?

< OK >

Le post-confoguration est maintenant terminé.

Postconfiguration completed.  
Please connect to <https://wapt.mydomain.lan/> to access the server.

< OK >

Liste des options du script de post-configuration :

Options	Description
<code>--force-https</code>	Configurer <b>Nginx</b> de sorte que <i>le port 80 soit redirigé de façon permanente vers le 443</i>

Le serveur est maintenant prêt. Vous pouvez consulter la documentation sur *l'installation de la console WAPT*.



---

## Installer le serveur WAPT sur CentOS7

---

### 11.1 Configurer le serveur CentOS / RedHat

Afin d'installer une nouvelle machine (virtuelle ou physique), veuillez vous référer à la documentation officielle de CentOS. Cette documentation est également valable pour Redhat7.

**Avertissement :**

- Installez la version **64bit**.
- Installez le serveur sans interface graphique.

**Danger :** Pour le Serveur WAPT **Nginx** est le **SEUL** server web supporté. **Apache n'est plus supporté par WAPT.**

#### 11.1.1 Configurer les paramètres réseau

Les différents paramétrages préconisés ci-dessous ne sont pas spécifiques à WAPT ; vous devrez les adapter à votre environnement.

Modifier les fichiers suivant afin d'obtenir une configuration de nommage (*FQDN*) et réseau correcte (adressage IP fixe).

Dans l'exemple suivant :

- le *FQDN* est *srvwapt.mydomain.lan* ;
- le nom court du serveur WAPT est *srvwapt* ;
- le suffixe *DNS* est *mydomain.lan* ;
- l'adresse IP est *10.0.0.10/24* ;

## 11.1.2 Configurer le nom du serveur WAPT

**Indication :** Le nom court du serveur WAPT ne doit pas dépasser 15 caractères (limite liée au sAMAccountName dans Active Directory).

---

Le nom du serveur doit être un nom FQDN, c'est à dire à la fois le nom de machine et le suffixe DNS.

- Modifier le fichier `/etc/hostname` et y renseigner le nom *FQDN* du serveur.

```
# /etc/hostname of the waptserver
srvwapt.mydomain.lan
```

- Configurez le fichier `/etc/hosts`, assurez-vous de mettre à la fois le *FQDN* et le nom court du serveur.

```
# /etc/hosts of the waptserver
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.0.10   srvwapt.mydomain.lan srvwapt
```

**Indication :**

- Sur la ligne définissant l'adresse IP du serveur DNS, veillez à avoir l'IP du serveur (pas 127.0.0.1), puis le *FQDN*, puis le nom court.
  - Ne modifiez pas la ligne avec `localhost`.
- 

## 11.1.3 Configurer l'adresse IP du serveur WAPT

- Modifiez le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0` et définissez une adresse IP statique. Le nom du fichier peut être différent, comme `ifcfg-ens0` par exemple.

```
# /etc/sysconfig/network-scripts/ifcfg-eth0 of the WAPT server
TYPE="Ethernet"
BOOTPROTO="static"
NAME="eth0"
ONBOOT="yes"
IPADDR=10.0.0.10
NETMASK=255.255.255.0
GATEWAY=10.0.0.254
DNS1=10.0.0.1
DNS2=10.0.0.2
```

- Appliquez la configuration réseau en redémarrant la machine avec un `reboot`.

```
reboot
```

- Si ce n'est pas déjà fait, *créer les entrées DNS pour le serveur WAPT* dans le *Organisation* Active Directory ou sur votre serveur DNS.
- Après le redémarrage, configurez la langue du système en anglais afin d'avoir des journaux non localisés pour faciliter la recherche des erreurs courantes.

```
localectl set-locale LANG=en_US.utf8
localectl status
```

— Vérifiez que l'horloge de la machine est à l'heure (avec NTP installé), et que SELinux et le pare-feu sont activés.

```
yum list installed | grep ntp
service ntpd status
date
sestatus
systemctl status firewalld
```

**Indication :** Si le paquet NTP n'est pas installé.

```
yum install ntp -y
systemctl enable ntpd.service
systemctl start ntpd
```

— Mettre à jour CentOS et configurez le dépôt EPEL (Extra Packages for Enterprise Linux).

```
yum update
yum install epel-release wget sudo -y
```

Le serveur est maintenant prêt. Vous pouvez maintenant passer à l'étape suivante et *installer WAPT sur votre CentOS/ RedHat*.

**Attention :** La procédure est différente pour la mise à jour du serveur WAPT. Pour une mise à jour, rendez-vous sur *la documentation pour mettre à jour le serveur WAPT*.

L'installation de la partie serveur de WAPT se décompose en plusieurs étapes :

- Configurer les dépôts.
- Installation de paquets Linux supplémentaires.
- Installation et provisionnement de la base de données PostgreSQL.
- Post-configuration du serveur WAPT.

## 11.2 Configuration du dépôt RPM

La configuration des dépôts en **WAPT Enterprise** et en **WAPT Discovery** diffère. **Assurez-vous de choisir la bonne procédure!!**

### 11.2.1 Discovery

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition **WAPT Discovery**. Pour **WAPT Enterprise** Edition, veuillez vous référer au bloc suivant.

---

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera publié ultérieurement. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>

---

### 11.2.2 Enterprise

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition **WAPT Enterprise**. Pour **WAPT Discovery** Edition, veuillez vous référer au bloc précédent.

Pour accéder au site de téléchargement **WAPT Enterprise**, vous devez utiliser le nom d'utilisateur et le mot de passe fournis par notre service commercial.

---

— Ajout du dépôt Tranquil'IT.

---

**Indication :** Remplacez **user** et **password** pour accéder au référentiel **WAPT Enterprise**, par ceux fournis par notre service commercial.

---

```
cat > /etc/yum.repos.d/wapt.repo <<EOF
[wapt]
name=WAPT Server Repo
baseurl=https://srvwapt-pro.tranquil.it/entreprise/centos7/wapt-2.1/
username=user
password=password
enabled=1
gpgcheck=1
EOF
```



## 11.3 Installer des paquets complémentaires

— Récupération de la clé .gpg.

```
wget -q -O /tmp/tranquil_it.gpg "https://wapt.tranquil.it/centos7/RPM-GPG-KEY-TISWAPT-7"; rpm --
→import /tmp/tranquil_it.gpg
```

— Installez tous les paquets nécessaires.

```
yum install epel-release -y
yum install postgresql96-server postgresql96-contrib tis-waptserver tis-waptsetup cabextract -y
```

— Initialiser la base de données PostgreSQL et activer les services.

```
sudo /usr/pgsql-9.6/bin/postgresql96-setup initdb
sudo systemctl enable postgresql-9.6 waptserver nginx
sudo systemctl start postgresql-9.6 nginx
```

## 11.4 Post-configuration

**Attention :** Pour que la post-configuration fonctionne correctement, vous devez d'abord avoir un *hostname* pour le serveur WAPT. Pour vérifier, utilisez la commande **echo \$(hostname)** qui doit retourner l'adresse DNS qui sera utilisé par les agents WAPT sur les ordinateurs clients.

**Avertissement :** Le script de post-configuration réécrit la configuration de nginx.

Si vous utilisez une *configuration spéciale*, sauvegardez votre fichier `wapt.conf` avec la commande

```
cp /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-available/wapt.conf.old
```

Il sera nécessaire de réécrire la configuration après la post-configuration avec la commande :

```
cp /etc/nginx/sites-available/wapt.conf.old /etc/nginx/sites-available/wapt.conf
```

**Indication :** Ce script de post-configuration doit être exécuté en tant que **root**.

— Lancez le script.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

— Cliquez sur *Oui* pour lancer le script de post-configuration.

```
do you want to launch post configuration tool?
```

```
< yes >          < no >
```

- Choisissez un mot de passe (si ce n'est pas déjà défini) pour le compte *SuperAdmin* du serveur WAPT (longueur minimale de 10 caractères).

```
Please enter the wapt server password (min. 10 characters)

*****

                < OK >          < Cancel >
```

- Confirmer le mot de passe.

```
Please enter the server password again:

*****

                < OK >          < Cancel >
```

- Choisissez le mode d'authentification pour l'enregistrement initial des agents WAPT :
  - Le choix #1 permet d'enregistrer les ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistré.
  - Le choix #2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer plus tard).
  - Le choix #3 n'active pas le mécanisme d'authentification kerberos pour l'enregistrement initial des machines équipées de WAPT. Le serveur WAPT va demander un identifiant et mot de passe pour chaque machine qui s'enregistre.

```
WaptAgent Authentication type?

-----
(x) 1 Allow unauthenticated registration
( ) 2 Enable kerberos authentication required for machines registration.
    Registration will ask for password if kerberos not available
( ) 3 Disable kerberos but registration require strong authentication
-----

                < OK >          < Cancel >
```

- Sélectionnez *OK* pour démarrer le serveur WAPT.

```
Press OK to start waptserver

                < OK >
```

- Sélectionnez *Oui* pour configurer Nginx.

```
Do you want to configure nginx?

                < Yes >          < No >
```

- Remplissez le *FQDN* du serveur WAPT.

```
FQDN for the WAPT server (eg. wapt.example.com)

-----
wapt.mydomain.lan
```

(suite sur la page suivante)

(suite de la page précédente)

```
-----
< OK >      < Cancel >
```

— Sélectionnez *OK* et un certificat auto-signé sera généré, cette étape peut prendre un certain temps.

```
Generating DH parameters, 2048 bit long safe prime, generator 2 This is going to take a long time
.....+.....+...
```

Nginx est maintenant configuré, sélectionnez *OK* pour redémarrer **Nginx** :

```
The Nginx config is done.
We need to restart Nginx?

< OK >
```

Le post-confoguration est maintenant terminé.

```
Postconfiguration completed.
Please connect to https://wapt.mydomain.lan/ to access the server.

< OK >
```

Liste des options du script de post-configuration :

Options	Description
--force-https	Configurer <b>Nginx</b> de sorte que <i>le port 80 soit redirigé de façon permanente vers le 443</i>

Votre serveur WAPT est maintenant prêt. Vous pouvez consulter la documentation sur *l'installation de la console WAPT*.



---

## Installer le serveur WAPT sur Windows

---

**Attention :**

- Le serveur WAPT ne peut pas être installé sur une machine qui écoute déjà sur le port 80 et 443 (exemple WSUS avec IIS).
- Les ports 80, 443 et 8080 sont utilisés par le serveur WAPT et doivent être disponibles.
- Si les ports 80 et 443 sont déjà occupés par un autre service web, vous devriez consulter la documentation officielle de Microsoft pour modifier les ports par défaut sous Windows.
- Le serveur WAPT **ne fonctionnera pas** sur une version x86 de Windows.
- L'installation du serveur WAPT doit être effectuée en utilisant un compte **Administrateur local** sur l'hôte et **PAS un compte Administrateur de domaine**.

**Danger :** **Nginx** est le **seul** serveur web supporté avec WAPT. **Apache** ou **IIS** (avec ou sans WSUS) **ne sont PAS supportés** par WAPT.

En cas de difficulté lors de l'installation de WAPT, visitez *la Foire Aux Questions*.

**Note :**

- L'installation de WAPT sur un serveur Linux est la méthode recommandée, sauf si vous testez WAPT et que vous n'êtes pas familier avec Linux.
- Le serveur WAPT peut être installé sur un **système 64bit seulement**, pour les nouvelles installations utiliser une .

**Indication :** Le composant serveur de WAPT fonctionne aussi bien sur une VM client win10 ou une machine physique que sur une version serveur de Windows.

## 12.1 Discovery

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

---

**Note :** Non disponible en date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>

---

## 12.2 Enterprise

**Indication :** Pour accéder aux ressources de WAPT **Enterprise**, vous devez utiliser le nom d'utilisateur et le mot de passe fournis par notre service commercial.

---

— Téléchargez et exécutez `waptserversetup.exe`.

**Attention :** L'installation du serveur WAPT doit être effectuée en utilisant un compte **Administrateur local** sur l'hôte et **PAS un compte Administrateur de domaine**.

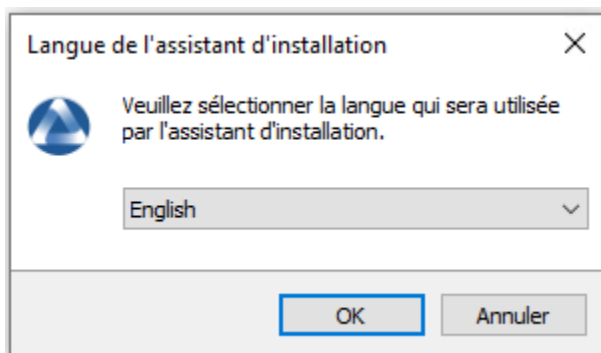
**Avertissement :** Le script de post-configuration réécrit la configuration de nginx. Si vous utilisez une configuration spéciale, sauvegardez votre fichier `nginx.conf` avec la commande :

```
copy C:\wapt\waptserver\nnginx\conf\nnginx.conf C:\wapt\waptserver\nnginx\conf\nnginx.conf.old
```

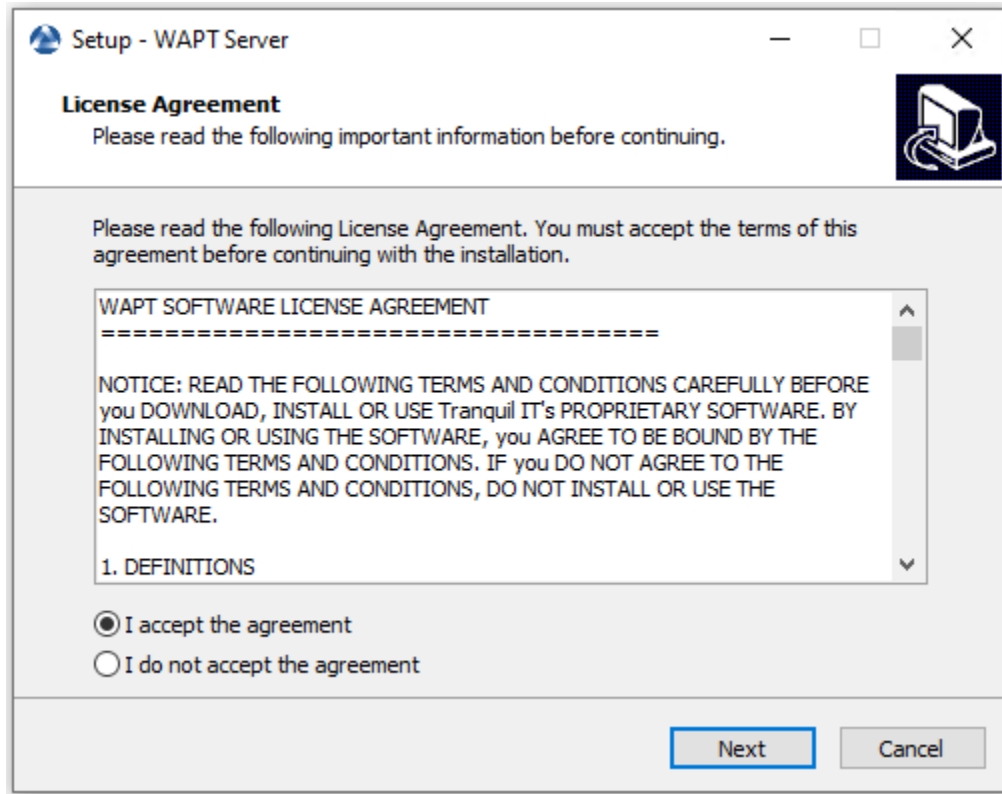
Il sera nécessaire d'écraser la configuration après le post-configuraiton avec la commande :

```
copy C:\wapt\waptserver\nnginx\conf\nnginx.conf.old C:\wapt\waptserver\nnginx\conf\nnginx.conf
```

— Choisissez la langue d'installation.



- Acceptez la licence publique GNU et cliquez sur *Suivant* pour passer à l'étape suivante.

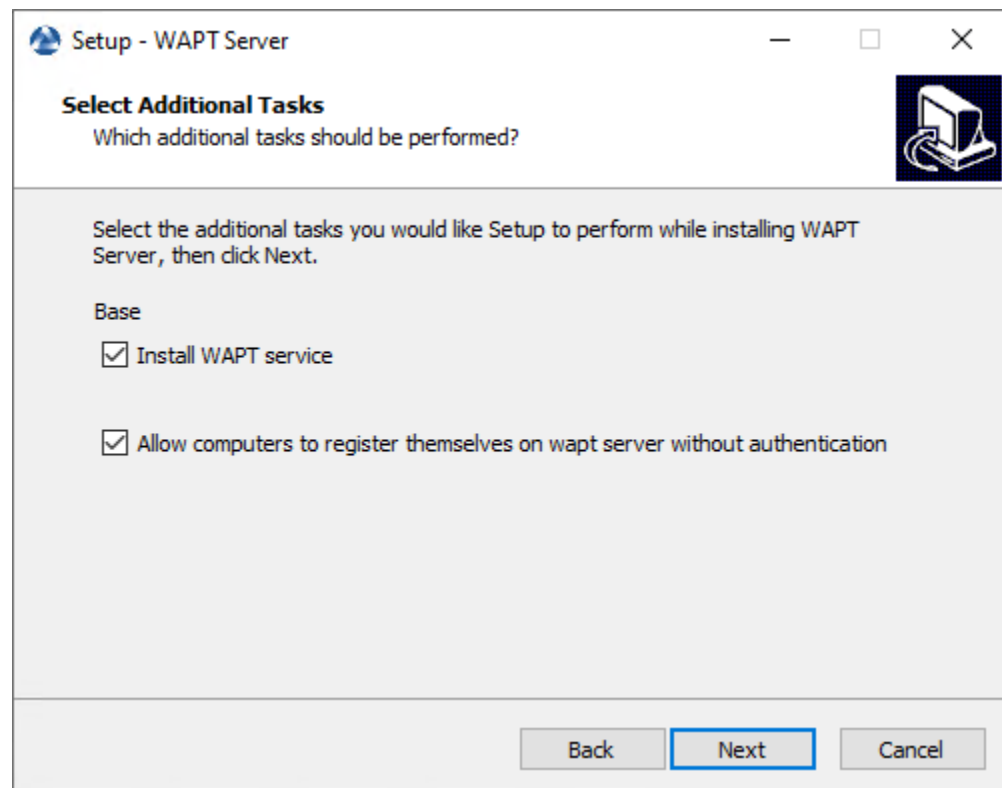
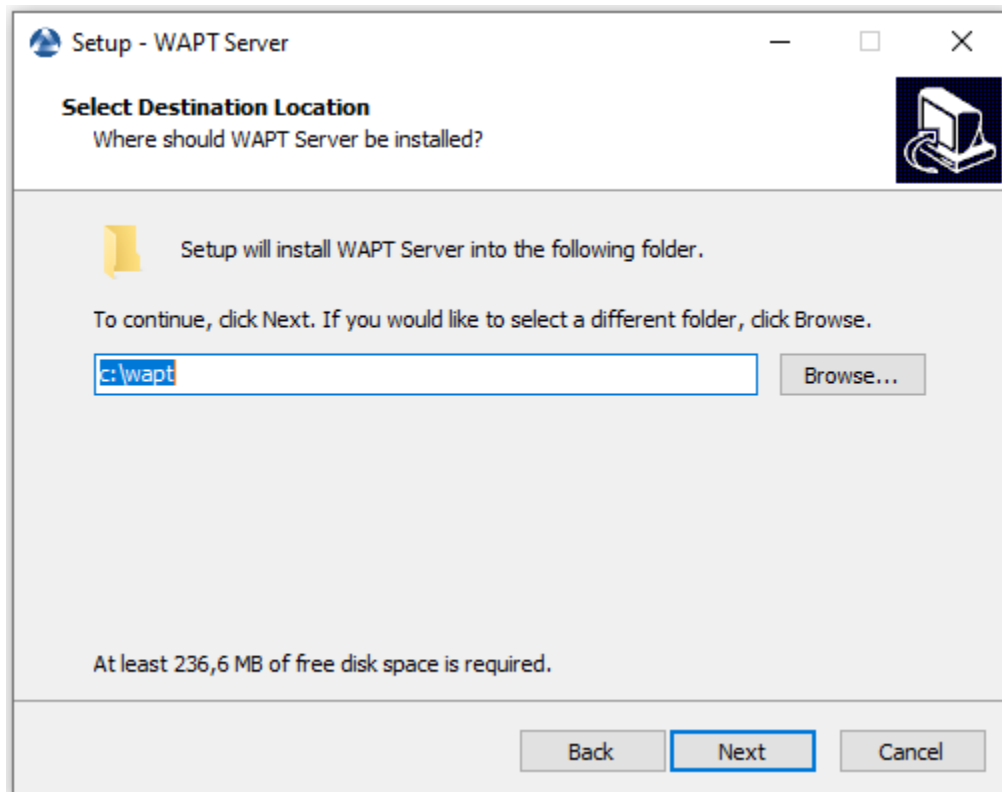


- Choisissez le répertoire d'installation (laissez celui par défaut) et cliquez sur *Next* pour passer à l'étape suivante.
- Choisissez une tâche supplémentaire (laissez la valeur par défaut si vous n'êtes pas sûr).
- Choisissez le mot de passe pour le serveur WAPT.
- Cliquez sur le bouton *Install* pour lancer l'installation, attendez que l'installation soit terminée.
- Cliquez sur *Terminé* pour fermer la fenêtre.

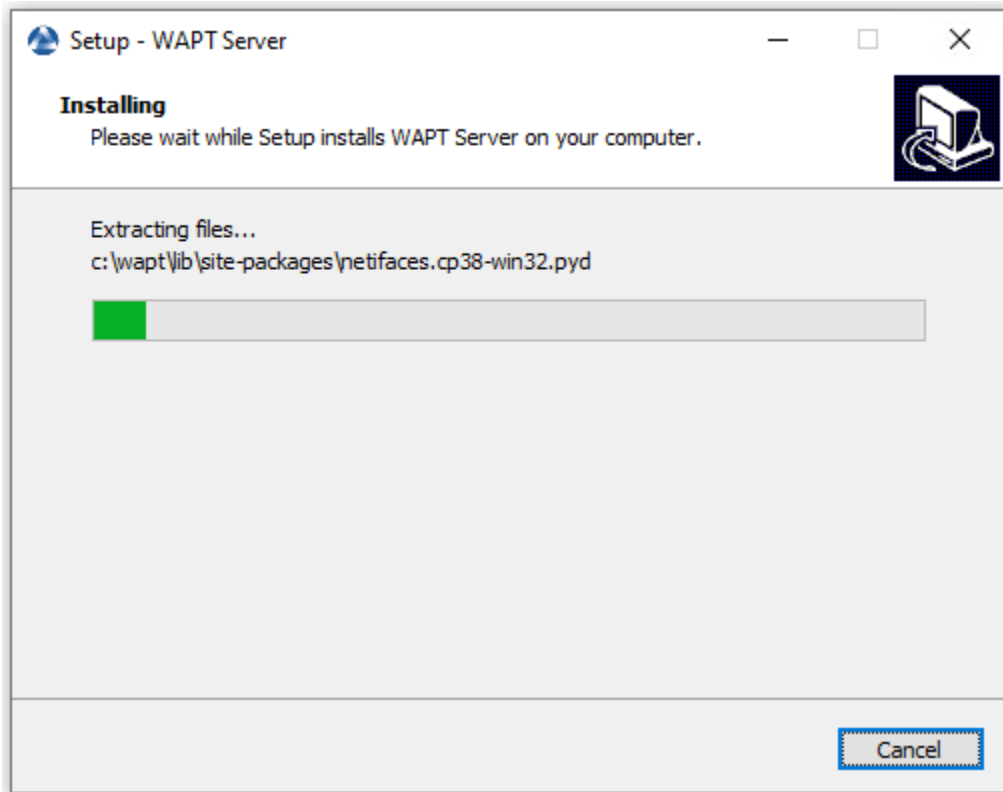
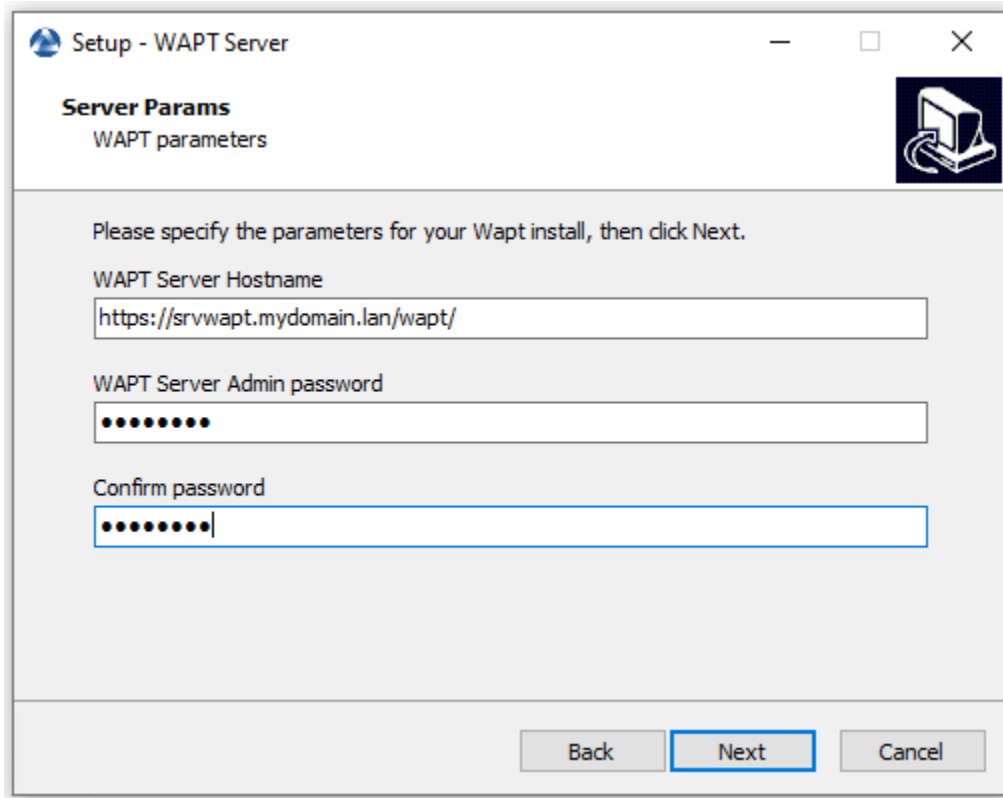
**Attention :**

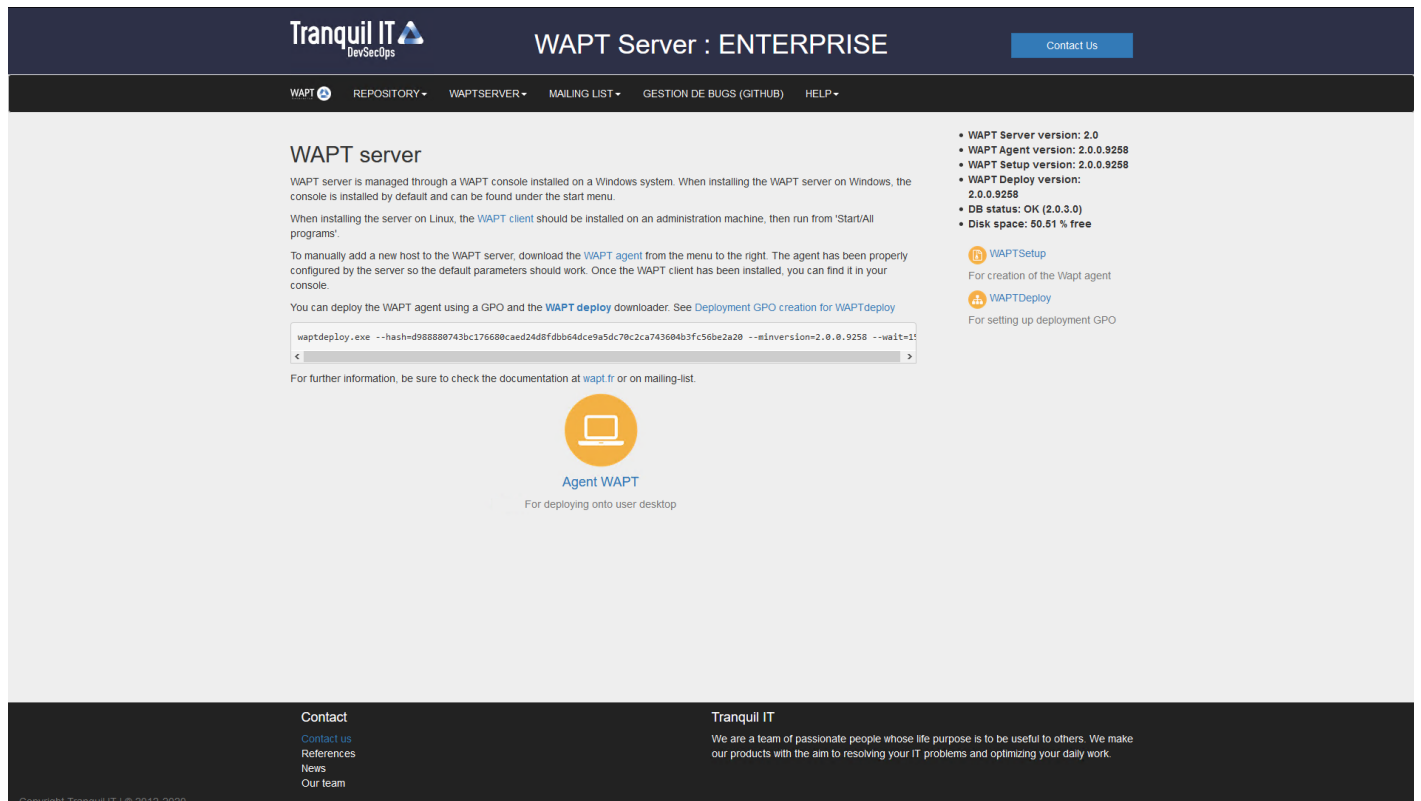
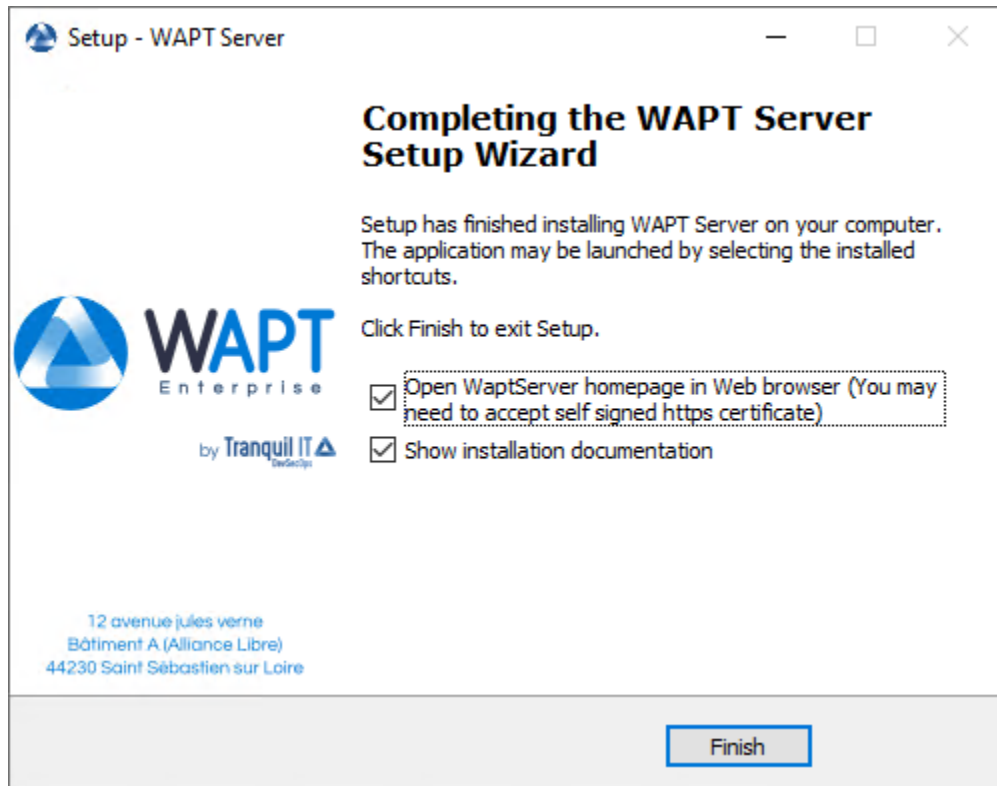
- **Pour des raisons de sécurité, n'exécutez pas la console WAPT ou votre outil de développement de paquet WAPT sur le serveur WAPT.**
- Le Serveur WAPT sur Windows **embarque un Agent WAPT**. Il n'est pas nécessaire d'installer l'Agent WAPT pour administrer le Server WAPT sur Windows.

Votre serveur est maintenant prêt. Vous pouvez maintenant consulter la documentation sur *Installation de la console de gestion WAPT*. Le serveur WAPT ayant été installé avec succès, nous allons maintenant installer la console WAPT.









---

## Console de gestion WAPT

---

**Attention :** Si vous avez déjà généré l'agent WAPT et déployé l'agent sur le poste de travail de votre *Administrateur*, alors lancez la console WAPT.

---

**Note :**

- La gestion de WAPT se fait principalement via la console WAPT installée sur le poste de travail de l'*Administrateur*.
- Il est recommandé de joindre l'ordinateur de l'administrateur à l'Active Directory de l'*Organisation*.
- Le nom d'hôte du poste de travail de l'administrateur **ne doit pas comporter plus de 15 caractères**, ce qui est une limite de l'attribut *sAMAccountName* dans Active Directory.
- **L'ordinateur de l'administrateur deviendra essentiel pour l'administration de WAPT et le test des paquets WAPT.**
- Si les enregistrements DNS sont correctement configurés, vous devriez être en mesure d'accéder à l'interface web WAPT en visitant <https://srvwapt.mydomain.lan>.
- A la date du 2024-01-09 la console WAPT ne s'installe que sous Windows.

---

**Indication :** Il est **hautement recommandé** d'utiliser la console sur une **machine de gestion dédiée**.

---

## 13.1 Si le serveur WAPT est installé sur un hôte Windows

**Avertissement :** La console WAPT **NE DOIT PAS** être installée sur votre serveur WAPT basé sur Windows.

La console WAPT doit être installée sur le poste de travail à partir duquel vous gérez votre réseau.

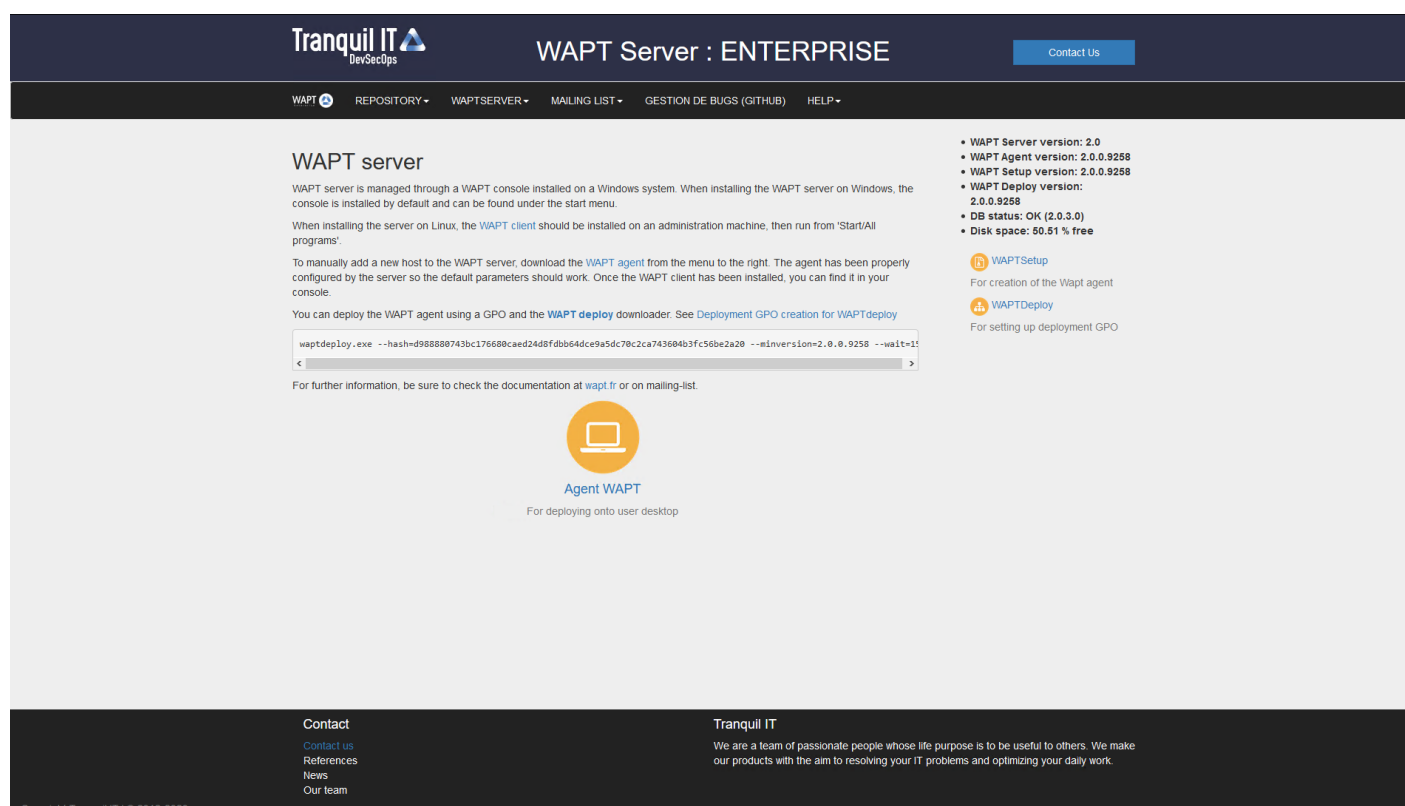
Pour installer la console WAPT, téléchargez `waptsetup.exe` sur le serveur de Tranquil IT vers le serveur WAPT.

- Renommez le fichier `waptsetup-tis.exe`.
- Copiez-le dans `C:\wapt\waptserver\repository\wapt`.

Vous pouvez maintenant poursuivre *le téléchargement et le lancement de l'installation de la console WAPT sur l'ordinateur de l'Administrateur*

## 13.2 Si le serveur WAPT est installé sur un hôte Linux

Passez à l'étape suivante, la Console WAPT est déjà sur votre serveur.



The screenshot displays the WAPT Server : ENTERPRISE web interface. At the top, there is a navigation bar with the Tranquil IT logo and a 'Contact Us' button. Below the navigation bar, there are several menu items: WAPT, REPOSITORY, WAPTSERVER, MAILING LIST, GESTION DE BUGS (GITHUB), and HELP. The main content area is titled 'WAPT server' and contains several paragraphs of text explaining the server's management and installation. On the right side, there is a status panel with a list of system information: WAPT Server version: 2.0, WAPT Agent version: 2.0.0.9258, WAPT Setup version: 2.0.0.9258, WAPT Deploy version: 2.0.0.9258, DB status: OK (2.0.3.0), and Disk space: 50.51 % free. Below this, there are two buttons: 'WAPTSetup' for creating the WAPT agent and 'WAPTDeploy' for setting up the deployment GPO. At the bottom of the page, there is a footer with contact information and a copyright notice for Tranquil IT.

FIG. 1 – L'interface web du serveur WAPT

- Si les enregistrements DNS sont correctement configurés, vous devriez pouvoir accéder à l'interface web WAPT en vous rendant à l'adresse suivante : <https://srvwapt.mydomain.lan>.
- Cliquez sur le lien `WAPTSetup` sur le côté droit de la page web du serveur WAPT.

## 13.3 Installation sur l'ordinateur de l'administrateur

**Attention :** Si **waptagent** n'est pas compilé et installé sur votre ordinateur, vous devez installer waptsetup.

Sinon, la console WAPT est déjà installée avec le waptagent, il suffit de *la configurer*.

- Lancez le programme d'installation exécutable en tant que *Administrateur local* sur le poste de travail de l'*Administrateur*.
- Choisissez la langue et cliquez sur *OK* pour installer la console WAPT.

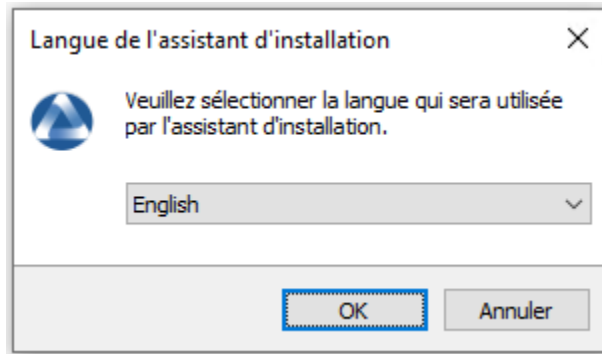


FIG. 2 – Choix de la langue pour WAPT

- Cliquez sur *OK* pour passer à l'étape suivante.
- Acceptez les conditions de la licence et cliquez sur *Next* pour passer à l'étape suivante.
- Choisissez vos options d'installation (les valeurs par défaut devraient convenir à la plupart des installations).

TABLEAU 1 – Choisissez les options de l'installateur

Paramètres	Description	Valeur par défaut
Installer le service WAPT	Ajoutez le service WAPT sur votre ordinateur de gestion.	Coché
Lancer l'icône de notification lors de l'ouverture de session	Lancer waptagent dans la barre d'état système au démarrage.	Non coché
Désactiver l'hiberboot, et augmenter le temps pour les GPO (recommandé)	Désactiver le démarrage rapide de Windows pour la stabilité, élargir le délai d'attente pour WAPTExit.	Coché
Installer les certificats fournis par cet installateur	Installez le certificat Tranquil iT uniquement sur cet ordinateur.	Non coché
Utiliser un UUID aléatoire pour identifier l'ordinateur au lieu du BIOS	Pour plus d'informations, consultez la documentation sur <i>BIOS UUID bugs</i>	Non coché

- Configurez l'URL du serveur WAPT .

**Indication :** Ici, deux choix s'offrent à vous.

- S'il s'agit de la première installation et que l'agent WAPT.
  - Vérifiez les « Informations statiques WAPT » et définissez-les :
    - URL du dépôt WAPT : <http://srvwapt.mydomain.lan/wapt>.
    - URL du serveur WAPT : <https://srvwapt.mydomain.lan>.
  - Choisir le dépôt et le serveur WAPT ; cliquer sur *Suivant*.

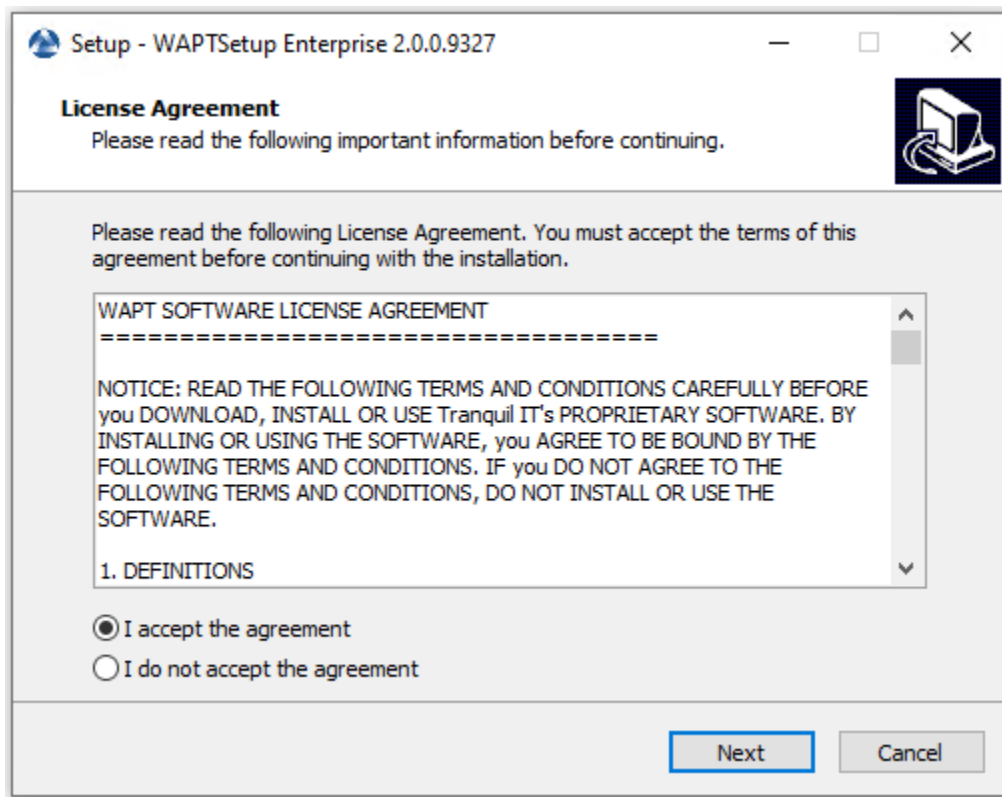


FIG. 3 – Acceptation des conditions de la licence WAPT

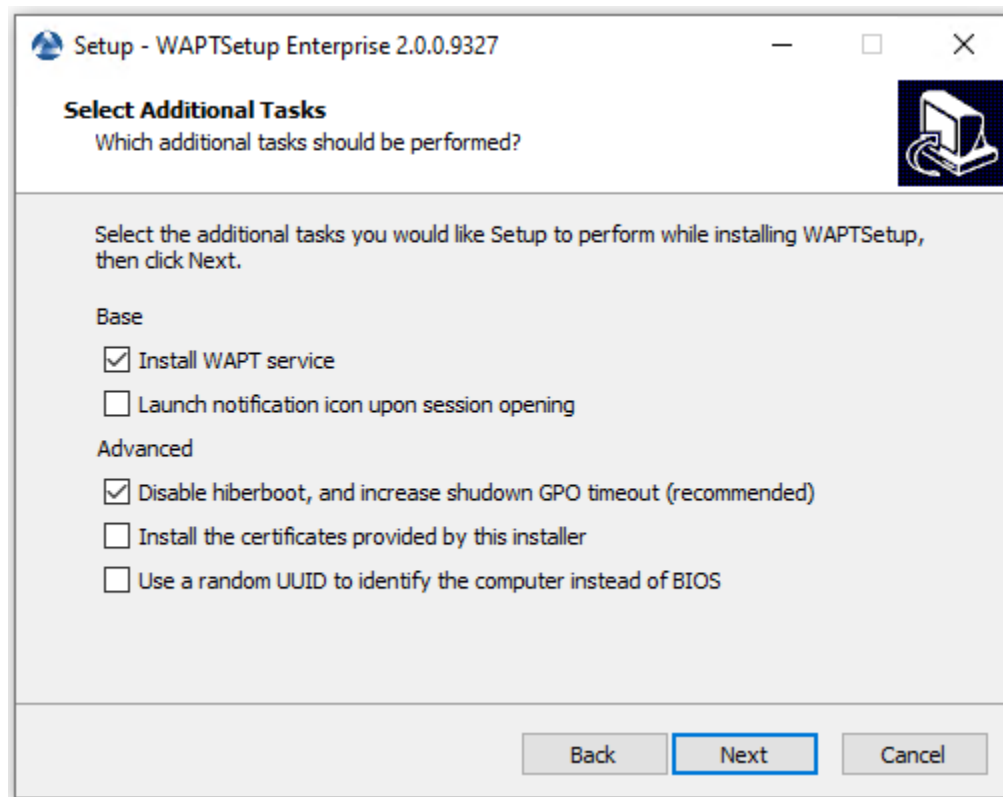


FIG. 4 – Choisir les options de l'installateur

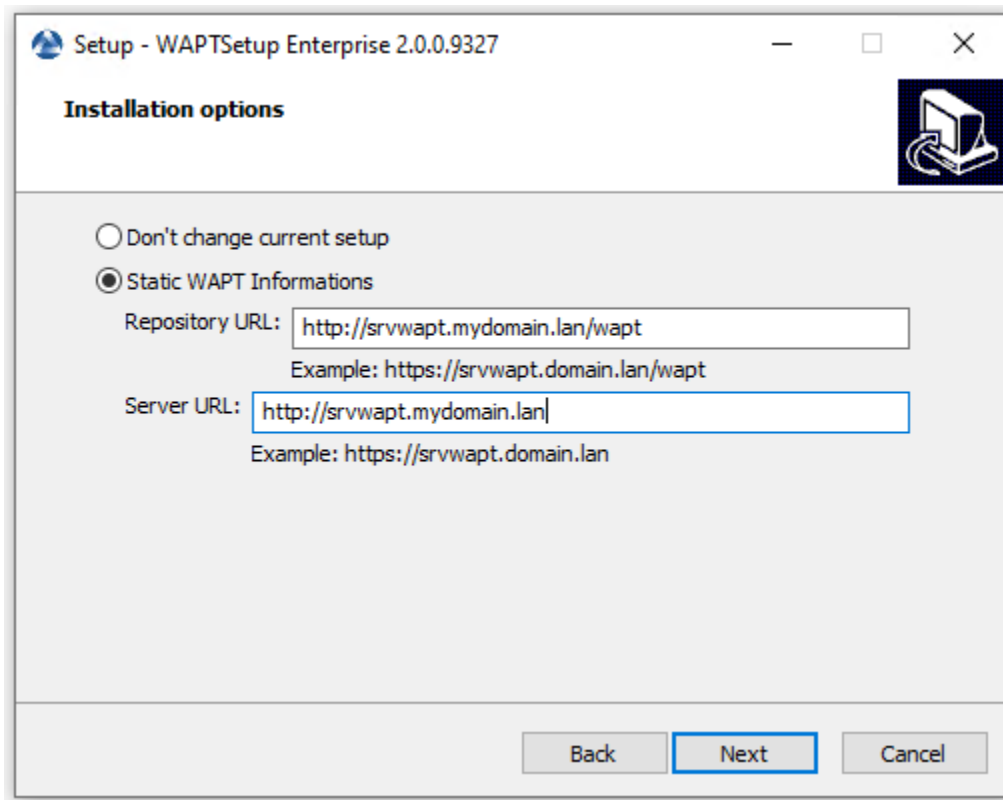


FIG. 5 – Choix du dépôt et du serveur WAPT



- Si la console WAPT ou l'agent WAPT est déjà installé :
  - Cochez *Ne pas modifier la configuration actuelle*, puis cliquez sur *Suivant*.

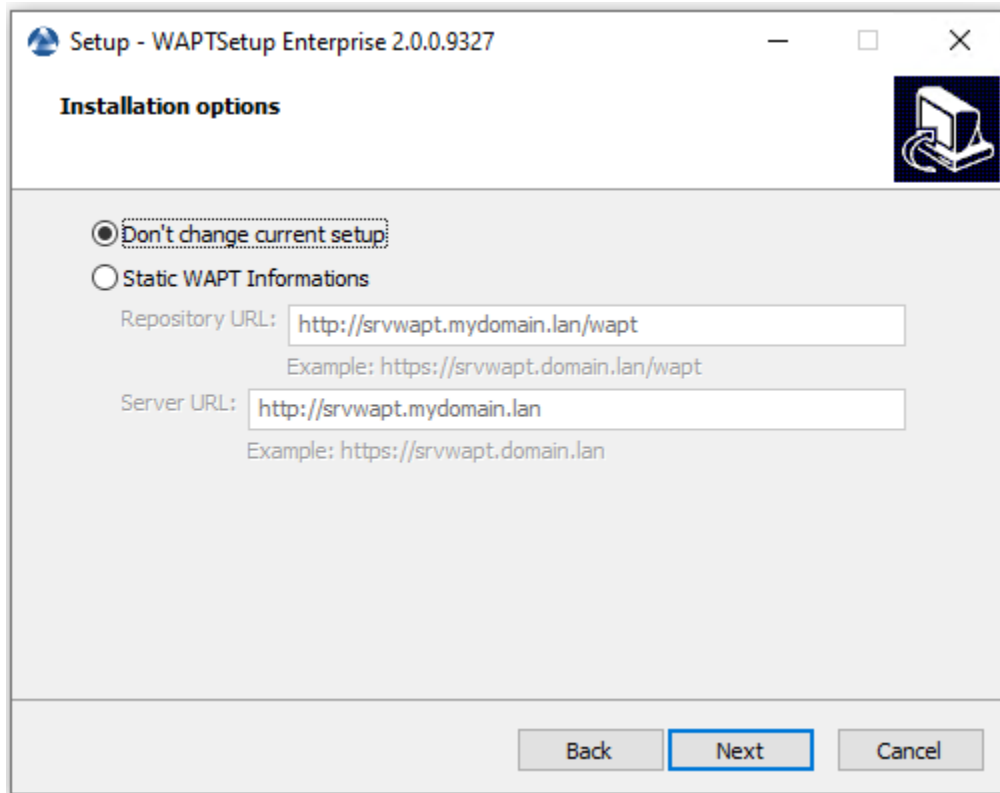


FIG. 6 – Le dépôt et le serveur WAPT sont déjà configurés

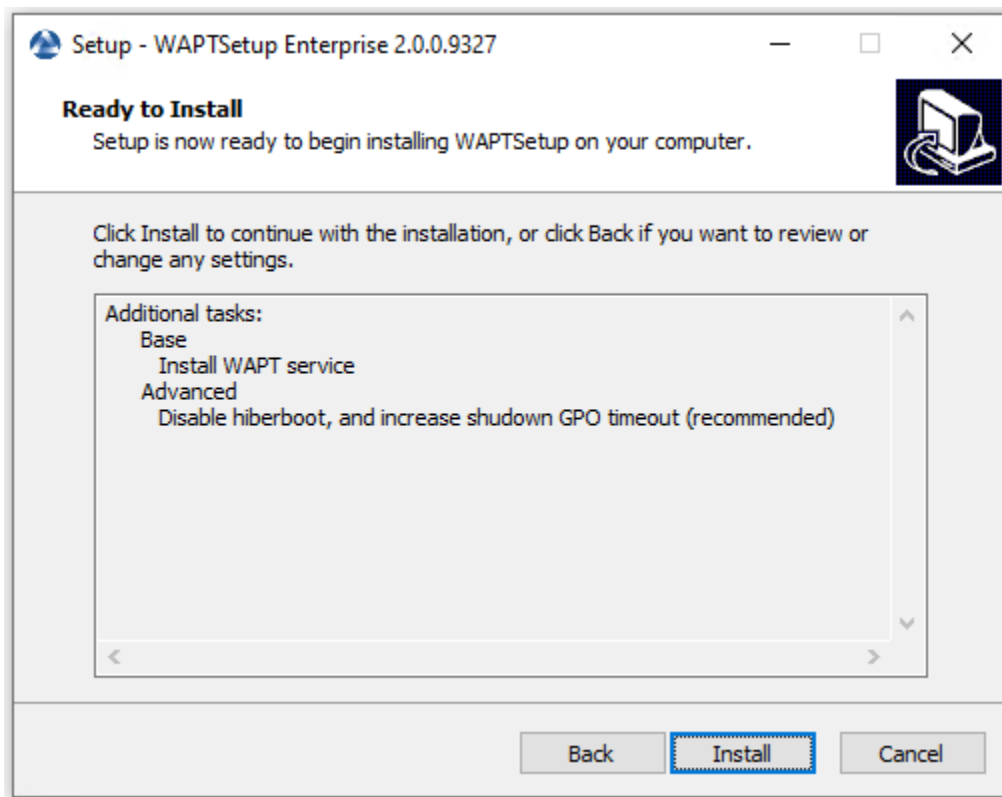
- Obtenir un résumé de l'installation de la console WAPT.
- Cliquez sur *Installer* pour lancer l'installation, attendez que l'installation se termine, puis cliquez sur *Terminé* (laissez les options par défaut).
- Décochez *Afficher la documentation d'installation*.

## 13.4 Démarrer la console WAPT

- Lancez la console WAPT :
  - En cherchant le binaire.  
C:\Program Files (x86)\wapt\waptconsole.exe
  - Ou en utilisant le menu *Démarrer*.
- Connectez-vous à la console WAPT avec le login et le mot de passe *SuperAdmin*.

Si vous avez des problèmes pour vous connecter à la console WAPT, veuillez vous référer à la FAQ : *Message d'erreur à l'ouverture de la console*.

Il est recommandé de lancer la console WAPT avec un compte d'administrateur local pour permettre le débogage local des paquets WAPT.



Pour la version Enterprise, il est possible de s'authentifier avec l'*Active Directory*.

### 13.4.1 Premier démarrage après l'installation du serveur

---

**Indication :** Au premier démarrage, vous devez lancer la console WAPT avec des privilèges élevés. *Cliquez avec le bouton droit de la souris sur le binaire de la console WAPT → Démarrer en tant qu'administrateur local.*

---

#### Affectation du certificat

---

**Note :** Un message peut apparaître indiquant qu'aucun certificat personnel n'a été défini.

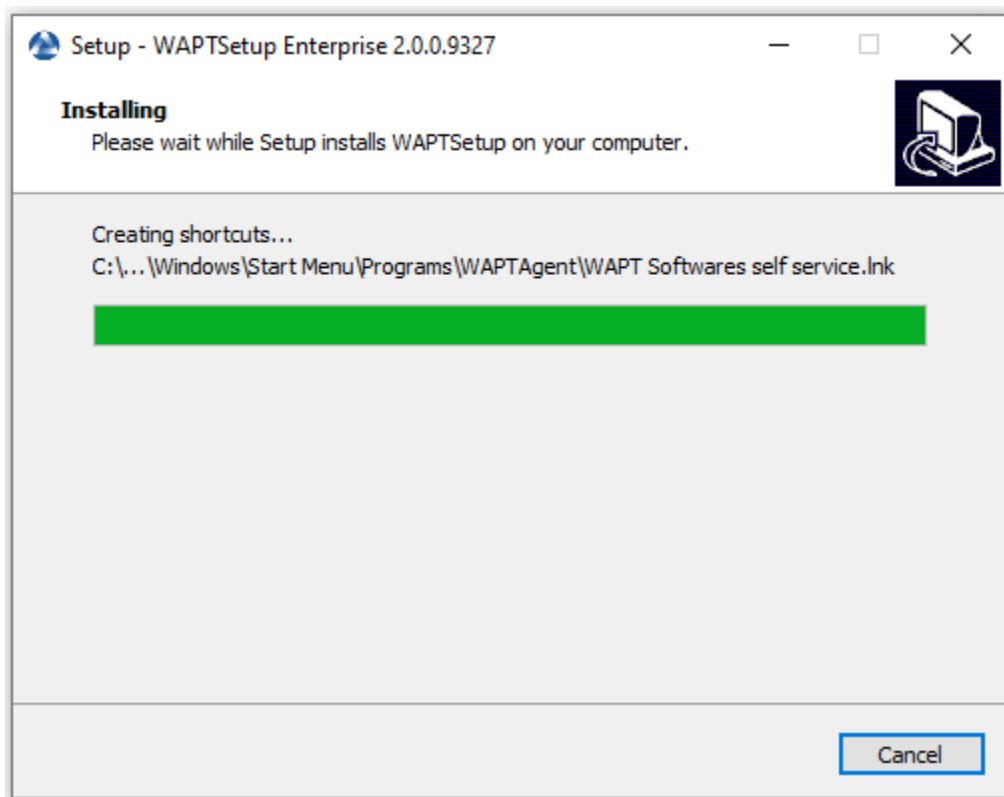


FIG. 7 – Assistant d'installation en cours

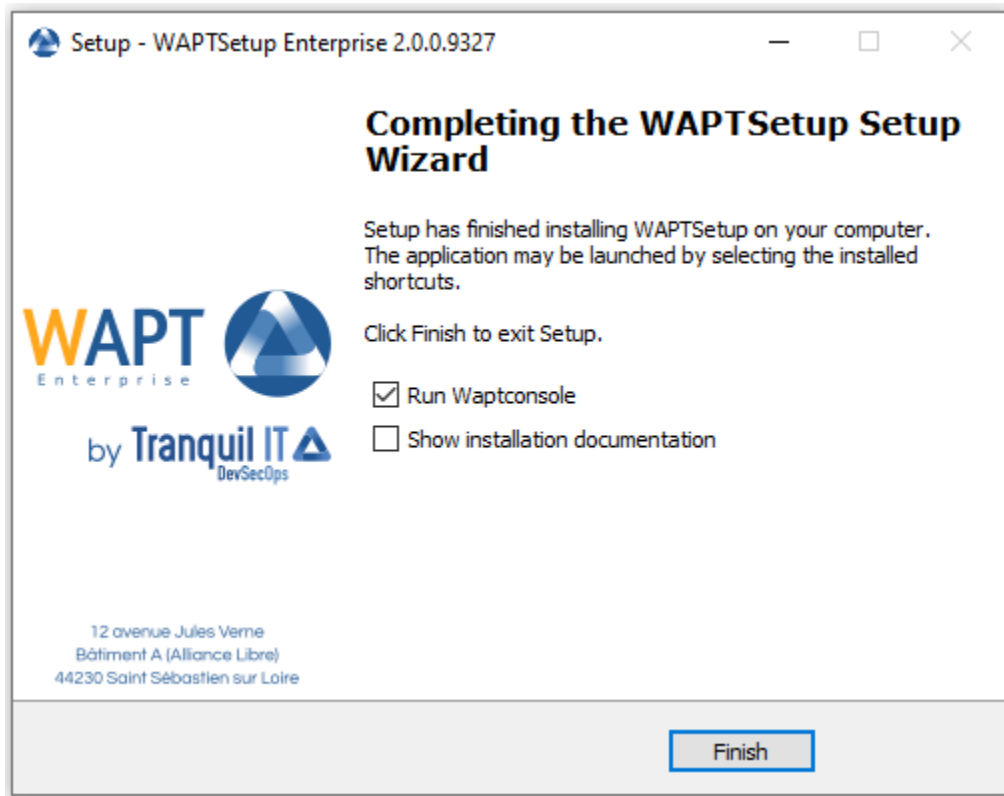


FIG. 8 – L’assistant d’installation est terminé

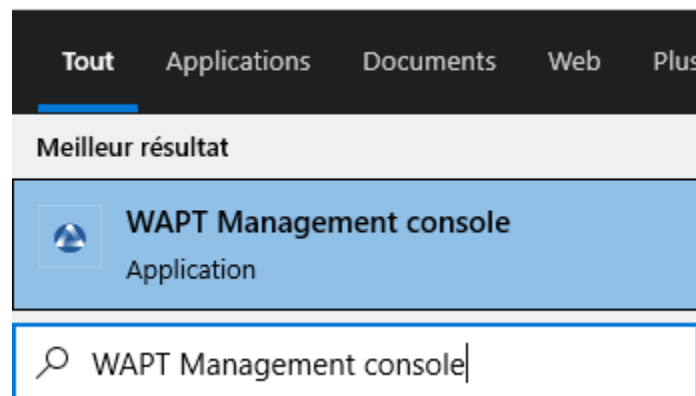


FIG. 9 – Le menu de démarrage de la console WAPT

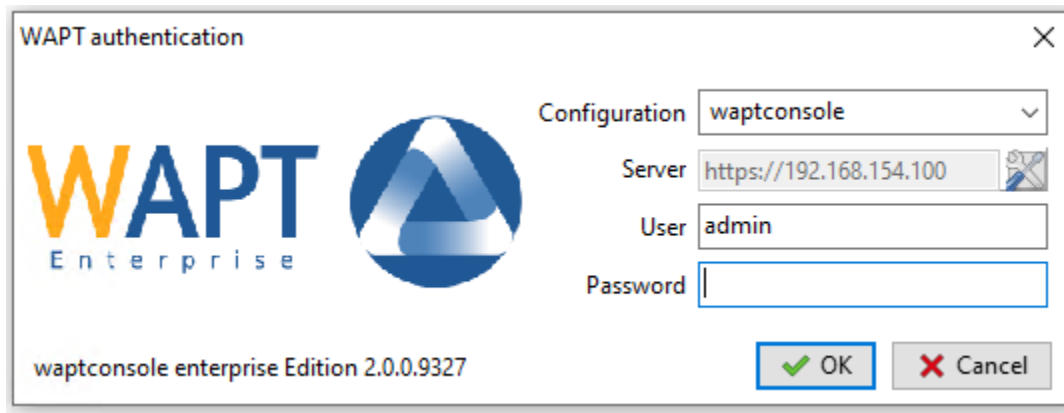
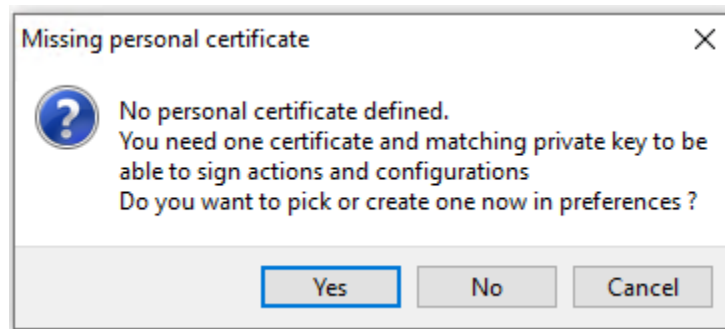


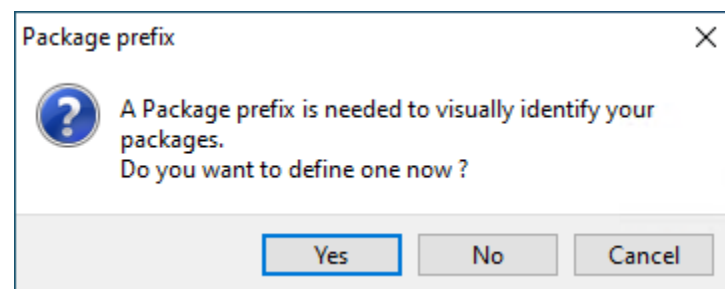
FIG. 10 – La fenêtre d'authentification de la console WAPT



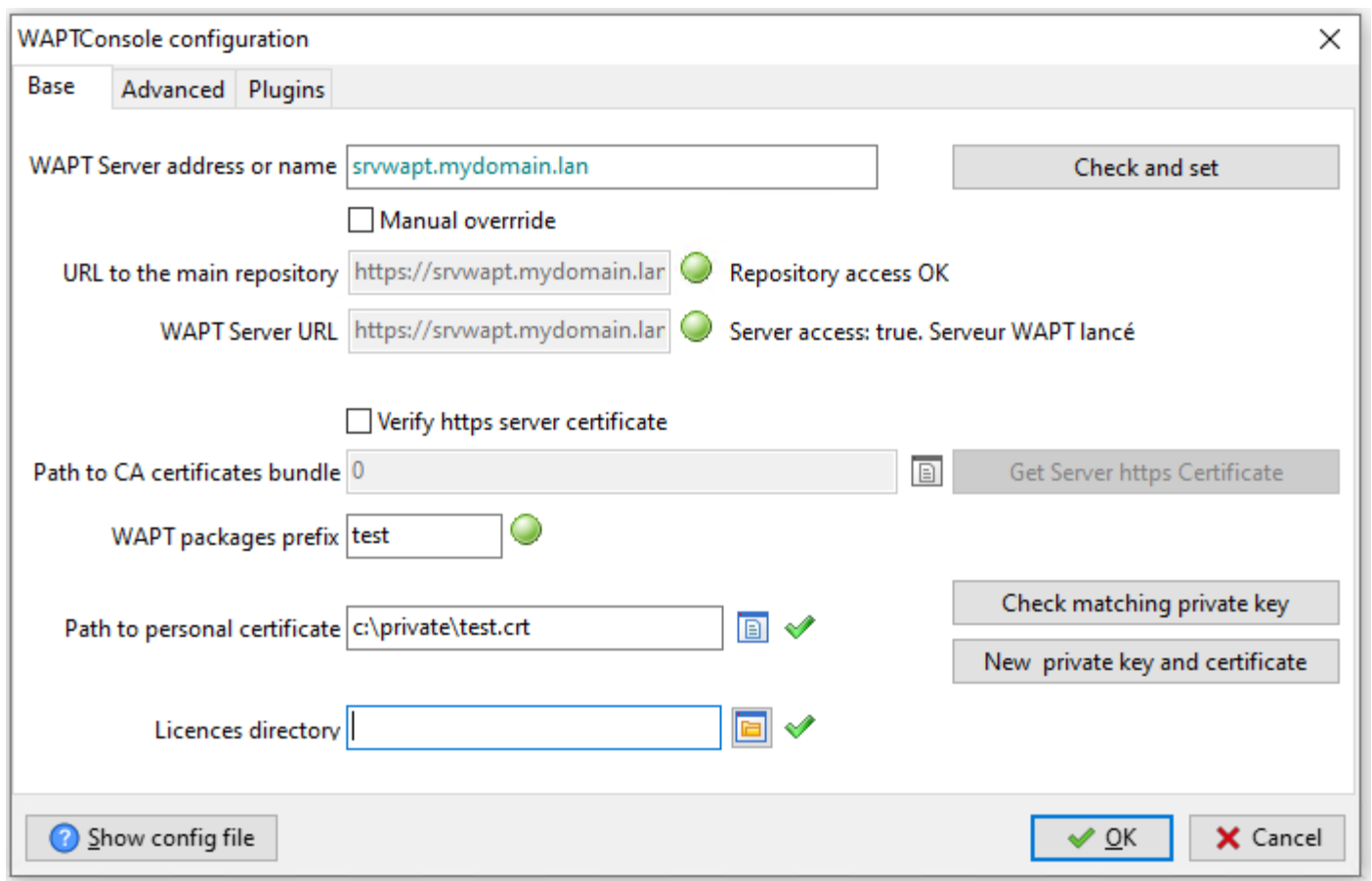
- Sélectionnez *Oui*
- Cliquez sur *Générer un certificat* puis allez créer votre certificat.

### Définition du préfixe de paquet

**Note :** Un message peut apparaître indiquant qu'aucun certificat personnel n'a été défini.



- Sélectionnez *Oui*



— Définissez votre préfixe de paquet sur *préfixe des paquets WAPT*

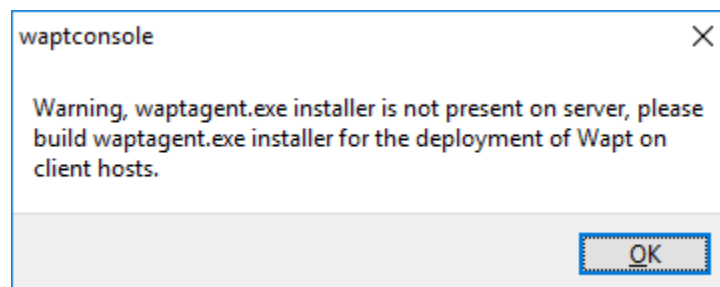
The screenshot shows the 'WAPTConsole configuration' dialog box with the 'Advanced' tab selected. The configuration includes:

- WAPT Server address or name:** `srvwapt.mydomain.lan` (with a 'Check and set' button).
- Manual override
- URL to the main repository:** `https://srvwapt.mydomain.lan` (with a green status indicator and text 'Repository access OK').
- WAPT Server URL:** `https://srvwapt.mydomain.lan` (with a green status indicator and text 'Server access: true. Serveur WAPT lancé').
- Verify https server certificate
- Path to CA certificates bundle:** `0` (with a folder icon and a 'Get Server https Certificate' button).
- WAPT packages prefix:** `test` (with a green status indicator).
- Path to personal certificate:** `c:\private\test.crt` (with a folder icon, a green checkmark, and buttons for 'Check matching private key' and 'New private key and certificate').
- Licences directory:** (empty field with a folder icon and a green checkmark).

At the bottom, there is a 'Show config file' button, an 'OK' button (highlighted with a blue border), and a 'Cancel' button.

## erreurs du waptagent.exe

**Note :** Un message peut apparaître indiquant que la version de votre agent WAPT est obsolète ou n'existe pas encore.



Si le *certicat de l'Administrateur* existe, il est possible de *générer un nouvel agent* en cliquant sur *Oui*.

Aussi, cliquez sur *Non* et générez le *certificat Administrateur*.





---

## Activation de la licence

---

---

**Note :** Sur WAPT, la différence entre les versions **Discovery** et **Enterprise** est gérée par la licence utilisée.

---

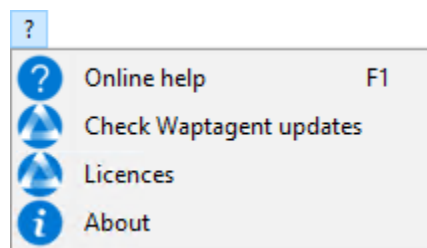
### 14.1 Activer la licence

---

**Indication :** Pour activer la licence, utilisez le fichier `licence.lic` communiqué par notre département de vente.

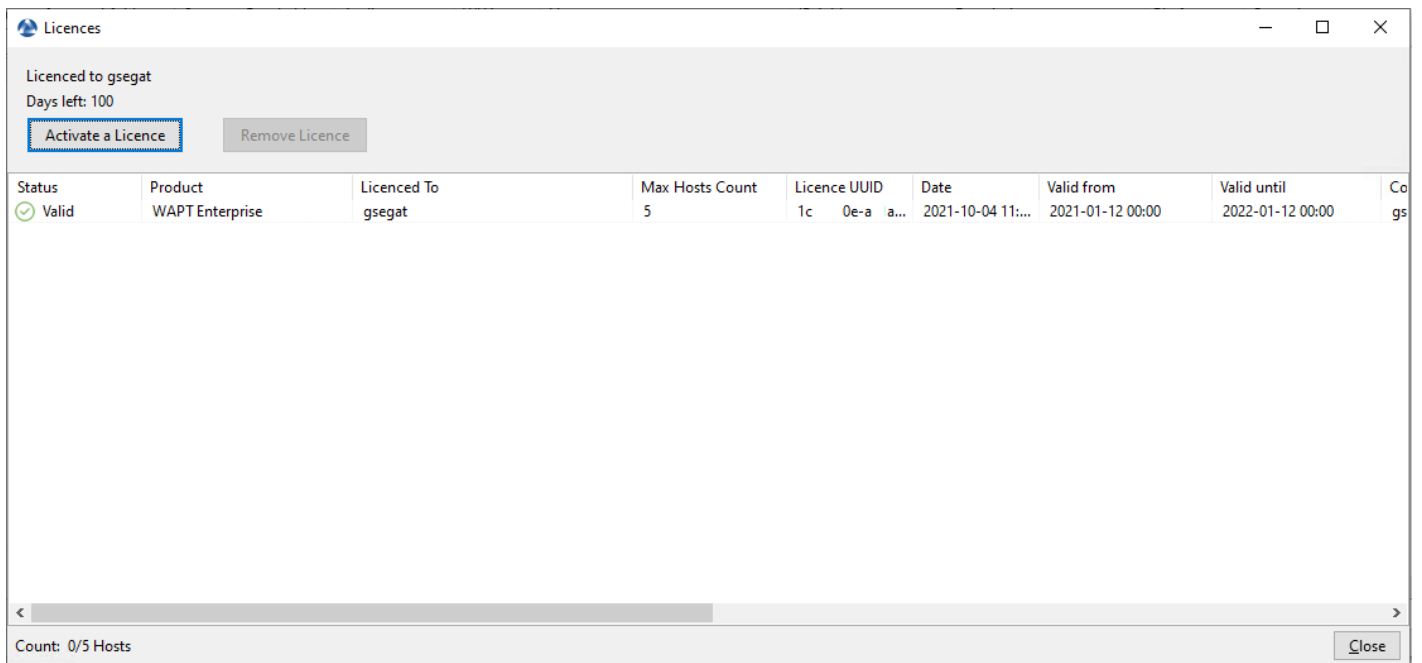
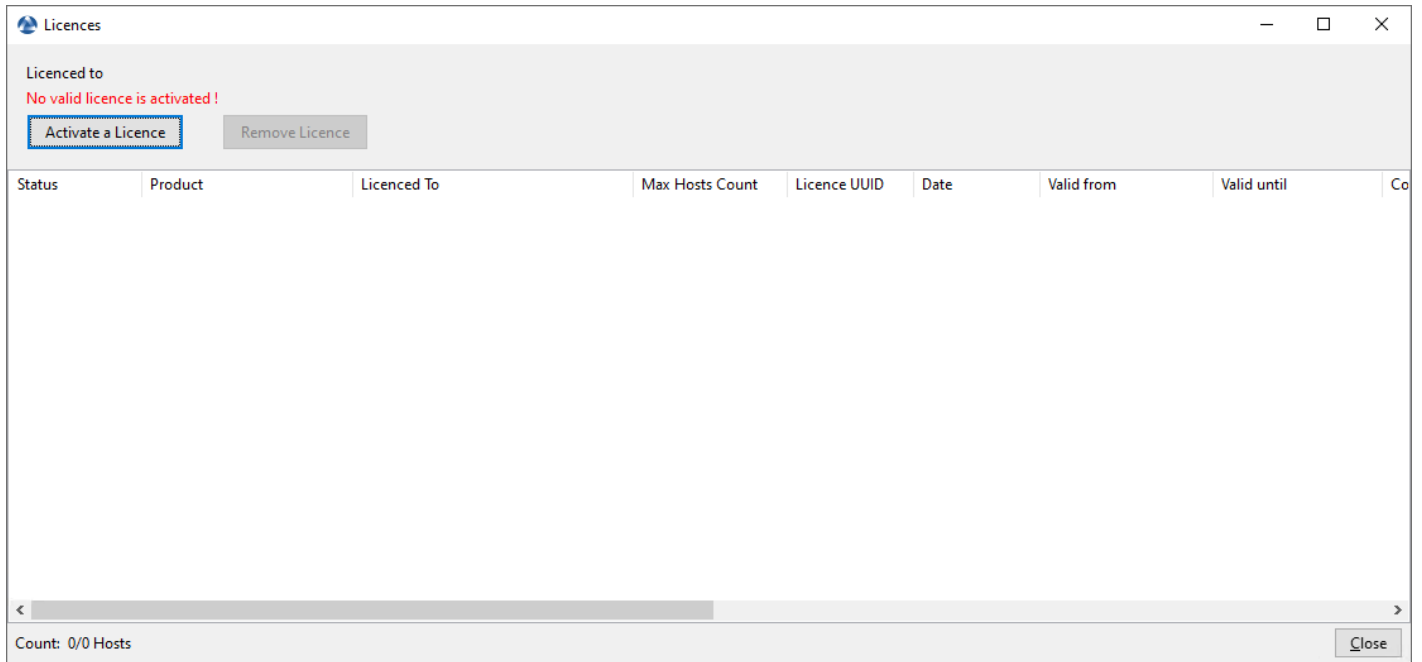
---

Dans la console WAPT, cliquez sur l'onglet ?



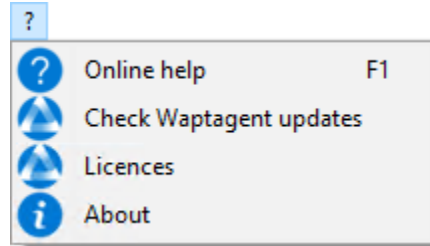
Choisissez *Licences* :

Sélectionnez votre `licence.lic` et cliquez sur *Ouvrir*.

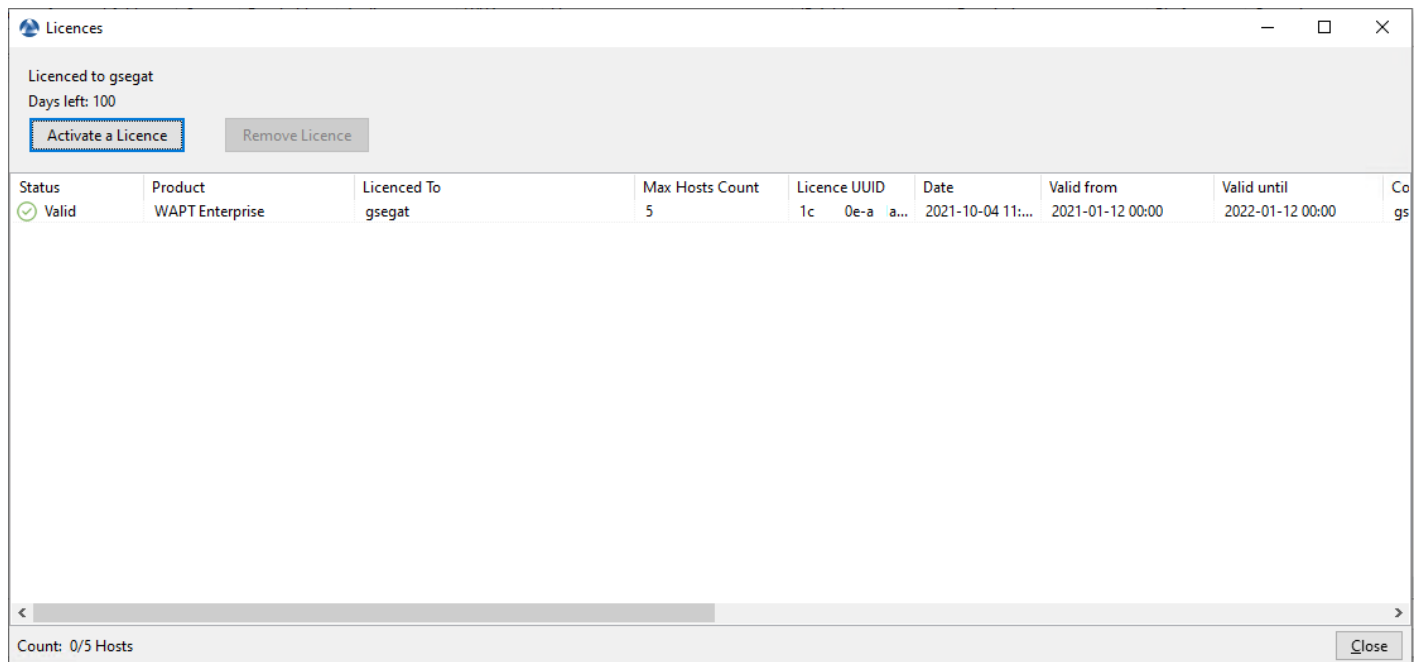


## 14.2 Supprimer la licence

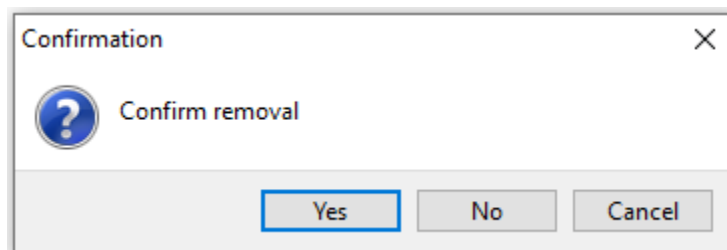
Dans la console WAPT, cliquez sur l'onglet ?



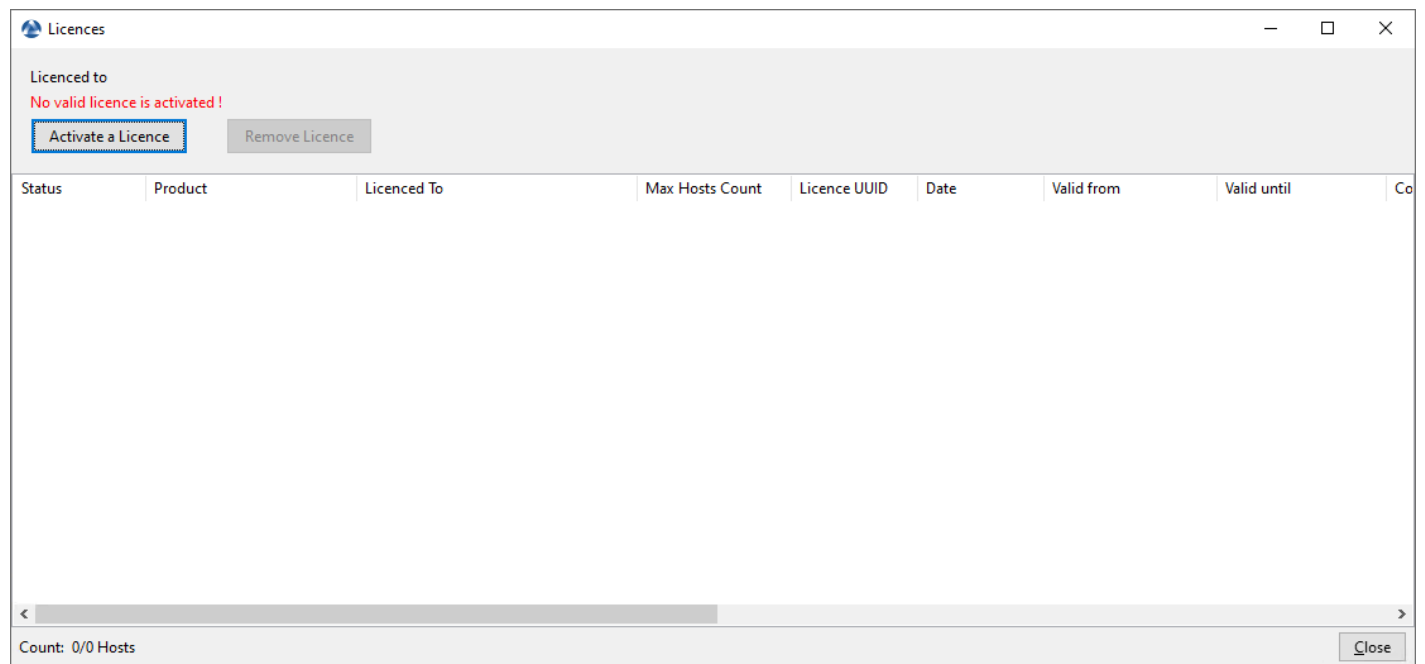
Choisissez *Licences* :



Sélectionnez la ligne et cliquez sur *Retirer la licence* :



Après confirmation, la licence est retirée :



## 14.3 Emplacement de la licence

Le `licence.json` est stocké sur le serveur sur l'emplacement suivant :

— Debian / Ubuntu :

```
/var/www/licences.json
```

— Red Hat / CentOS

```
/var/www/html/licences.json
```

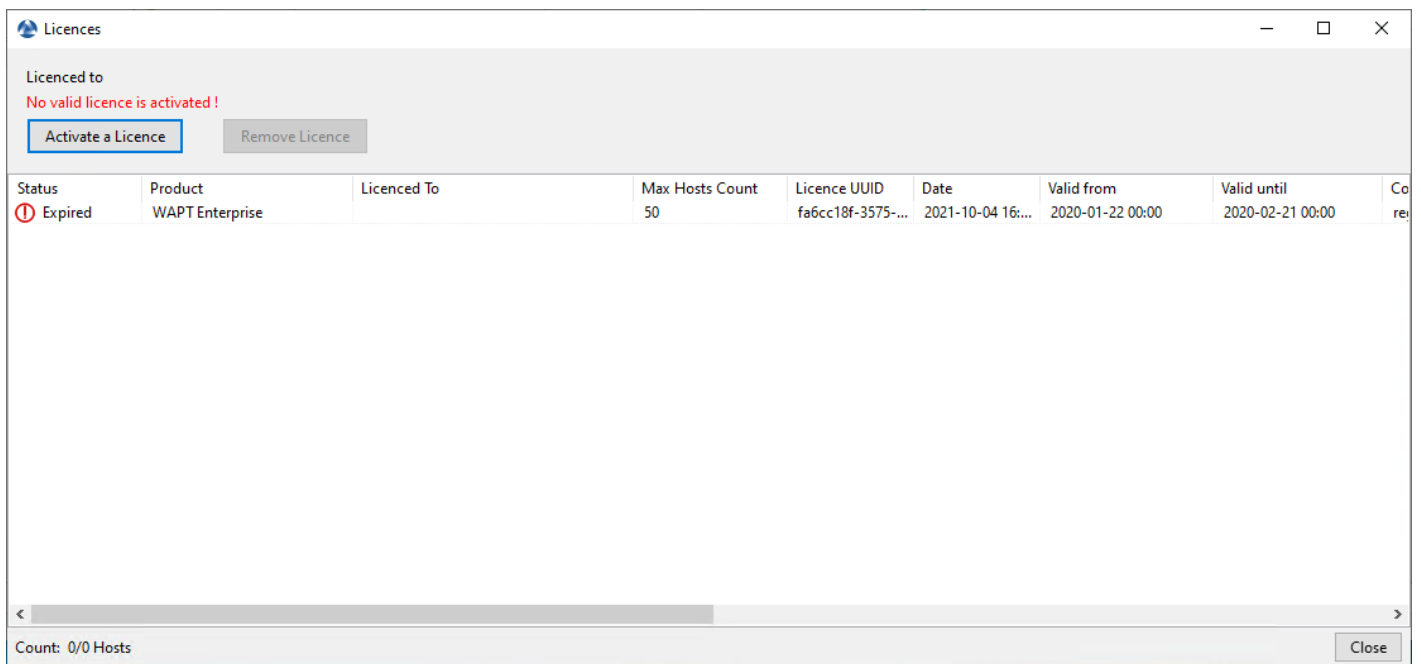
— Windows :

```
C:\wapt\waptserver\Repository\licences.json
```

## 14.4 Erreur de licence

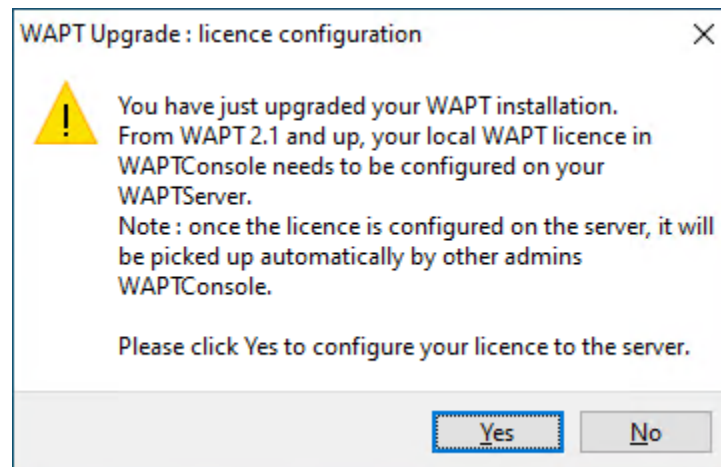
### 14.4.1 Expiration de la licence

Si la licence est expirée, le statut affiche *Expiré* :



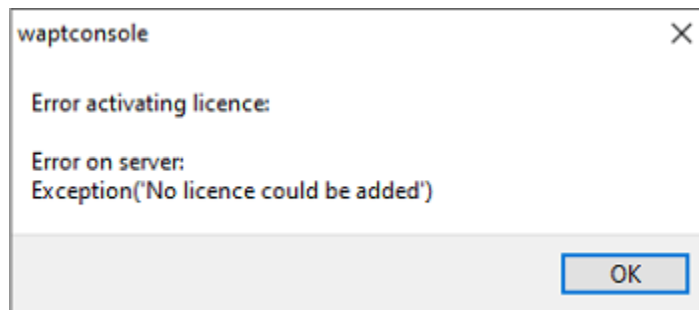
## 14.4.2 Emplacement de l'ancienne licence

Sur la console d'installation, si la licence est localisée sur l'ancien emplacement, cette erreur apparaît :



### 14.4.3 Erreur d'activation de la licence

Cette erreur est dû à un problème avec le script de post-configuration et une configuration spéciale de NGINX.



3 points sont à vérifier :

1. Vérifier si `/etc/nginx/sites-enabled/wapt.conf` est un lien symbolique du fichier `/etc/nginx/sites-available/wapt.conf`.

— Utilisez cette commande :

```
ls -l /etc/nginx/sites-enabled/wapt.conf
```

— Si le lien symbolique existe, la sortie est :

```
lrwxrwxrwx 1 root root 36 Jun  9 09:35 /etc/nginx/sites-enabled/wapt.conf -> /etc/nginx/sites-available/wapt.conf
```

— Egalement, supprimer le fichier et créer le lien symbolique :  
— supprimer `/etc/nginx/sites-enabled/wapt.conf`

```
rm /etc/nginx/sites-enabled/wapt.conf
```

— créer le lien symbolique :

```
ln -s /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-enabled/wapt.conf
```

2. Vérifier si `licences.json` est présent dans la section `location` du fichier `/etc/nginx/sites-enabled/wapt.conf`

```
location ~ ^/(wapt/waptsetup-tis.exe|wapt/waptagent.exe|wapt/waptdeploy.exe|sync.
->json|rules.json|licences.json)$ {
    add_header Cache-Control "store, no-cache, must-revalidate, post-check=0, pre-
->check=0";
    add_header Pragma "no-cache";
    root "/var/www";
}
```

— S'il existe, redémarrer NGINX :

```
systemctl restart nginx
```

— Egalement, ajouter `licences.json` dans la section `location` dans `/etc/nginx/sites-enabled/wapt.conf` et redémarrer NGINX.

```
systemctl restart nginx
```

3. Si l'erreur existe aussi :

— /var/www/licences.json vide :

```
> /var/www/licences.json
```

— Retenter *licence activation*





---

## Génération du certificat principal

---

---

### Indication :

- Le nom de la clé privée est `wapt-private.pem`.
  - Le nom du certificat public signé avec la clé privée est `wapt-private.crt`.
- 

### 15.1 Clef privée *wapt-private.pem*

**Attention :** Le fichier `wapt-private.pem` est **fondamental pour la sécurité**. Il doit être stocké dans un endroit sûr et correctement protégé.

Le fichier `wapt-private.pem` est la clé privée, il est situé par défaut dans le dossier `C:\private` du poste *Administrateur* et est protégé par un mot de passe.

Cette clé privée sera utilisée avec le certificat pour signer les paquets avant de les télécharger sur le dépôt WAPT.

**Danger :** Le fichier `wapt-private.pem` ne doit **PAS** être stocké sur le serveur WAPT.

## 15.2 Certificat public : *wapt-private.crt*

Le fichier `wapt-private.crt` est le certificat public qui est utilisé avec la clé privée. Il est créé par défaut dans le dossier `C:\private` de l'administrateur, copié et déployé dans `C:\Program Files (x86)\wapt\ssl` sur les postes de travail Windows ou dans `/opt/wapt/ssl` sur les périphériques Linux et MacOS gérés par l'administrateur via un package WAPT, un GPO ou un rôle Ansible.

Ce certificat est utilisé pour valider la signature des paquets avant leur installation.

**Attention :**

- Si le certificat public utilisé sur la console WAPT n'est pas dérivé de la clé privée utilisée pour générer les agents WAPT, aucune interaction ne sera possible.
- Les certificats enfants des clés privées sont fonctionnels pour les interactions.

## 15.3 Générer un certificat

Dans la console WAPT, allez dans *Outils* → *Générer un certificat*.

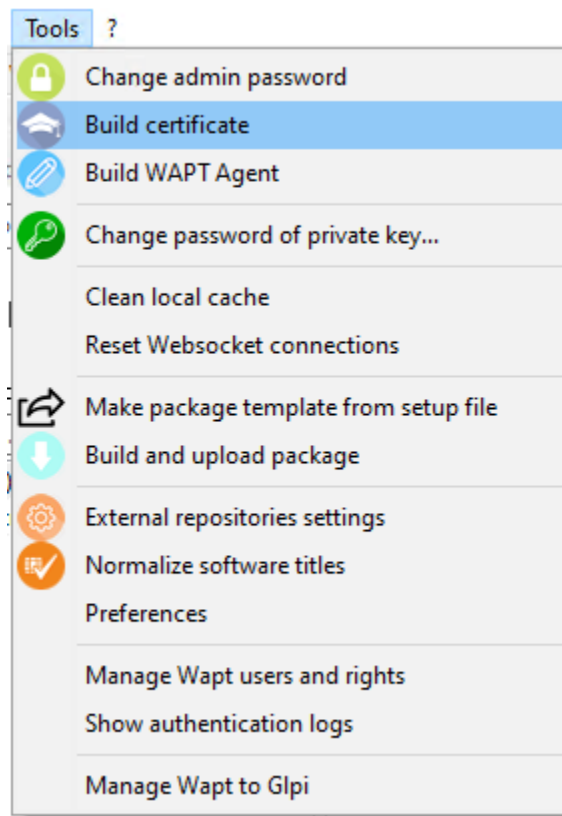


FIG. 1 – Générer un certificat auto-signé

---

**Important :** Nous avons deux options différentes :

- Créer un certificat pour la version Discovery.
- Créer un certificat pour la version Entreprise.

### 15.3.1 Discovery

- Remplissez les champs suivants.

Generate private key and self signed certificate

Target keys directory: C:\private

Key filename: C:\private\privatekey.pem

Private key password: \*\*\*\*\*

Confirm password: \*\*\*\*\*

---

Certificate name: privatekey

Common Name(CN): privatekey

**Optional information**

City:

Country (2 chars. E.g.: FR): FR

Service:

Organisation:

E-mail address:

Export PKCS12 too

OK Cancel

FIG. 2 – Créer un certificat pour la version Discovery

TABLEAU 1 – Informations sur le certificat

Valeur	Description	Requis
<i>Répertoire de destination des clés</i>	Dossier où seront stockés la clé privée et le certificat public.	✓
<i>Nom de fichier de la clé</i>	nom du <i>.pem</i> et <i>Nom de la clé privée</i> .	✓
<i>Mot de passe de la clé</i>	Mot de passe pour le verrouillage et le déverrouillage de la clé.	✓
<i>Confirmer le mot de passe de la clé</i>	Confirmation du mot de passe pour le verrouillage et le déverrouillage de la clé.	✓
<i>Nom du certificat</i>	Nom du certificat <i>.crt</i> .	✓
<i>Common Name (CN)</i>	Nom d’affichage du certificat.	✓
<i>Ville</i>	Nom de la ville du détenteur du certificat à enregistrer dans le certificat.	✗
<i>Pays (2 caractères. Exemple : FR)</i>	Nom du pays du détenteur du certificat (FR, EN, ES, DE ...) à enregistrer dans le certificat.	✗
<i>Service</i>	Nom du service ou du département du détenteur du certificat à enregistrer dans le certificat.	✗
<i>Organisation</i>	Nom de l’Organisation du détenteur du certificat à enregistrer dans le certificat.	✗
<i>Adresse E-Mail</i>	Adresse Mail du détenteur du certificat à enregistrer dans le certificat.	✗
<i>Export PKCS12</i>	Créer un certificat <i>*.p12</i> dans <i>Répertoire cible des clés</i> .	✗ (recommandé)

Des détails supplémentaires sont stockés dans la clé privée. Ces informations permettront d’identifier l’origine du certificat et l’origine du paquet WAPT.

**Indication :** La complexité du mot de passe doit être conforme aux exigences de sécurité de votre *Organisation* (visitez le site Web de l’ANSSI pour des recommandations sur les mots de passe).

**Danger :**

- Le chemin d’accès à votre clé privée ne doit pas se trouver dans le chemin d’installation de WAPT (C:\Program Files (x86)\wapt).
- Si votre clé est stockée dans C:\Program Files (x86)\wapt, votre clé privée *Administrateur* sera déployée sur vos clients, **absolument à proscrire!**
- Le fichier *wapt-private.pem* ne doit pas être stocké sur le serveur WAPT.

— Cliquez sur *OK* pour passer à l’étape suivante.

Si tout s’est bien passé, le message suivant apparaît :

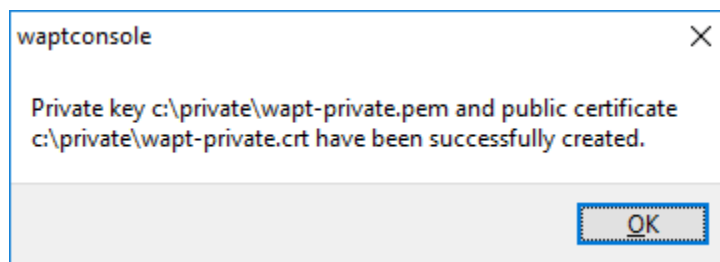


FIG. 3 – Le certificat a été généré avec succès

— Cliquez sur *OK*.

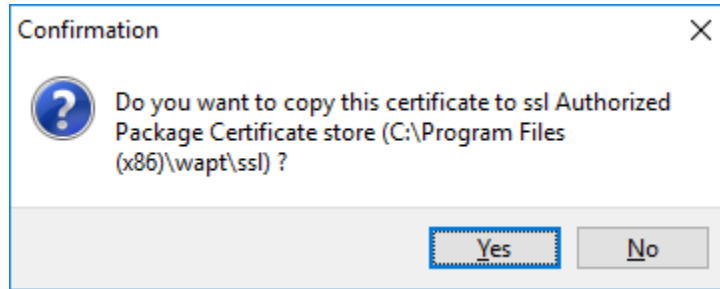


FIG. 4 – Confirmation de la copie du certificat dans le dossier ssl

— Cliquez sur *Oui* pour copier le certificat nouvellement généré dans le dossier `C:\Program Files (x86)\wapt\ssl` sous Windows ou `/opt/wapt/ssl` sous Linux ou MacOS. Ce certificat sera récupéré lors de la compilation de l'agent WAPT et déployé sur les ordinateurs clients.

Vous pouvez passer à l'étape suivante et *Construire le programme d'installation de l'agent WAPT*.

### 15.3.2 Enterprise

Avec WAPT Enterprise, vous pouvez créer une clé principale d'autorité de certification qui peut à la fois signer des paquets et signer de nouveaux certificats.

---

**Indication :** Afin de créer de nouveaux certificats signés pour les utilisateurs délégués, veuillez vous référer à *créer un nouveau certificat*.

---

Generate private key and self signed certificate

Target keys directory:

Key filename :

Private key password:

Confirm password:

---

Certificate name:

Tag as code signing

Tag as CA Certificate

Common Name(CN) :

**Optional information**

City :

Country (2 chars. E.g. : FR):

Service :

Organisation:

E-mail address :

---

Authority Signing Key:

Authority Signing Certificate:

*If you don't provide a CA Certificate and key, your certificate will be self-signed.*

Export PKCS12 too

FIG. 5 – Créer un certificat pour la version Entreprise

TABLEAU 2 – Informations sur le certificat

Valeur	Description	Requis
<i>Répertoire de destination des clés</i>	Dossier où seront stockés la clé privée et le certificat public.	✓
<i>Nom de fichier de la clé</i>	nom du <code>.pem</code> et <i>Nom de la clé privée</i> .	✓
<i>Mot de passe de la clé</i>	Mot de passe pour le verrouillage et le déverrouillage de la clé.	✓
<i>Confirmer le mot de passe de la clé</i>	Confirmation du mot de passe pour le verrouillage et le déverrouillage de la clé.	✓
<code>:guilabel</code> :Pour Signature de code	Cochez cette case si le certificat/la paire de clés sera autorisé(e) à signer des paquets logiciels.	✓
<i>Pour usage en tant que CA</i>	Cochez cette case si ce certificat peut être utilisé pour signer d'autres certificats (Autorité de Certification principale ou intermédiaire).	✓
<i>Nom du certificat</i>	Nom du certificat <code>.crt</code> .	✓
<i>Common Name (CN)</i>	Nom d'affichage du certificat.	✓
<i>Ville</i>	Nom de la ville du détenteur du certificat à enregistrer dans le certificat.	✗
<i>Pays (2 caractères. Exemple : FR)</i>	Nom du pays du détenteur du certificat (FR, EN, ES, DE ...) à enregistrer dans le certificat.	✗
<i>Service</i>	Nom du service ou du département du détenteur du certificat à enregistrer dans le certificat.	✗
<i>Organisation</i>	Nom de l'Organisation du détenteur du certificat à enregistrer dans le certificat.	✗
<i>Adresse E-Mail</i>	Adresse mail du détenteur du certificat à enregistrer dans le certificat	✗
<i>Clé privée de l'autorité</i>	Clé privée (* <code>.pem</code> ) de l'autorité de certification (CA)	✗
<i>Certificat de l'autorité</i>	Certificat (* <code>.crt</code> ) de l'autorité de certification (CA)	✗
<i>Export PKCS12</i>	Créer un certificat * <code>.p12</code> dans <i>répertoire de destination des clés</i>	✗ (recommandé)

Des détails supplémentaires sont stockés dans la clé privée. Ces informations permettront d'identifier l'origine du certificat et l'origine du paquet WAPT.

**Indication :** La complexité du mot de passe doit être conforme aux exigences de sécurité de votre *Organisation* (visitez le site Web de l'ANSSI pour des recommandations sur les mots de passe).

**Note :** Si votre organisation est déjà équipée d'une *Autorité de Certification (CA)*, vous devrez remplir le certificat et la clé dans les champs *Clé privée de l'autorité* et *Certificat de l'autorité*.

Cette procédure vous permet de générer de nouveaux certificats ou paires de clés avec ou sans la fonction **Code Signing**.

Pour créer une autorité de certification, allez à la section *générer l'autorité de certification (CA)*.

**Danger :**

- Le chemin d'accès à votre clé privée ne doit pas se trouver dans le chemin d'installation de WAPT (C:\Program Files (x86)\wapt).
- Si votre clé est stockée dans C:\Program Files (x86)\wapt, votre clé privée d'administrateur sera déployée sur vos clients, **ce qui est absolument à proscrire!**
- Le fichier `wapt-private.pem` ne doit pas être stocké sur le serveur WAPT.

Si tout s'est bien passé, le message suivant apparaît :

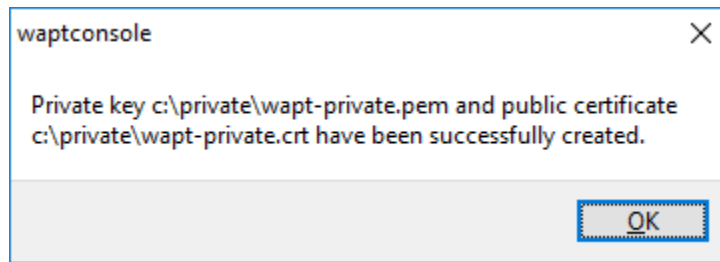


FIG. 6 – Le certificat a été généré avec succès

— Cliquez sur *OK* pour passer à l'étape suivante.

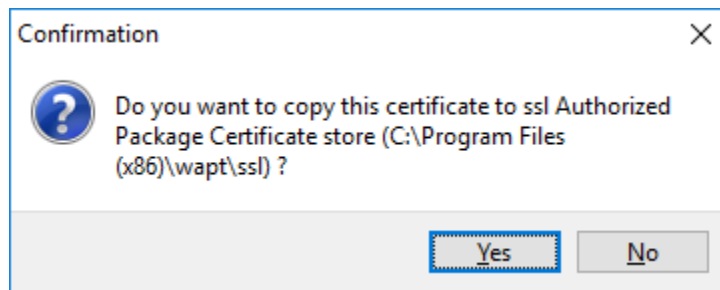


FIG. 7 – Confirmation de la copie du certificat dans le dossier ssl

— Cliquez sur *Oui* pour copier le certificat nouvellement généré dans le dossier C:\Program Files (x86)\wapt\ssl. Ce certificat sera récupéré lors de la compilation de l'agent WAPT et déployé sur les ordinateurs clients.

Vous pouvez passer à l'étape suivante et *construire le programme d'installation de l'agent WAPT*.



Le binaire **waptagent** est un installateur *InnoSetup*.

Une fois que la console WAPT a été installée sur l'ordinateur *Administrator*, nous avons tous les fichiers nécessaires pour construire le programme d'installation de l'agent WAPT :

- Les fichiers qui seront utilisés lors de la construction de l'agent WAPT sont situés dans `C:\Program Files (x86)\wapt`.
- Les fichiers sources du programme d'installation (fichiers `.iss`) se trouvent dans `C:\Program Files (x86)\wapt\waptsetup`.

---

**Indication :** Avant de construire l'agent WAPT, veuillez vérifier le(s) certificat(s) public(s) dans `C:\Program Files (x86)\wapt\ssl`.

Si vous souhaitez déployer d'autres certificats publics sur les ordinateurs de votre *Organisation* qui sont équipés de WAPT, vous devrez les copier dans ce dossier.

---

**Danger :** NE COPIEZ PAS la clé privée d'un *Administrator* dans `C:\Program Files (x86)\wapt`.

Ce dossier est utilisé lors de la construction de l'agent WAPT et les clés privées seront ensuite déployées sur tous les ordinateurs.

- Dans la console WAPT, allez dans *Outils* → *Générer un agent WAPT*

---

**Indication :** Avant de construire l'agent WAPT, vous devez choisir comment il s'identifiera auprès du serveur WAPT.

---

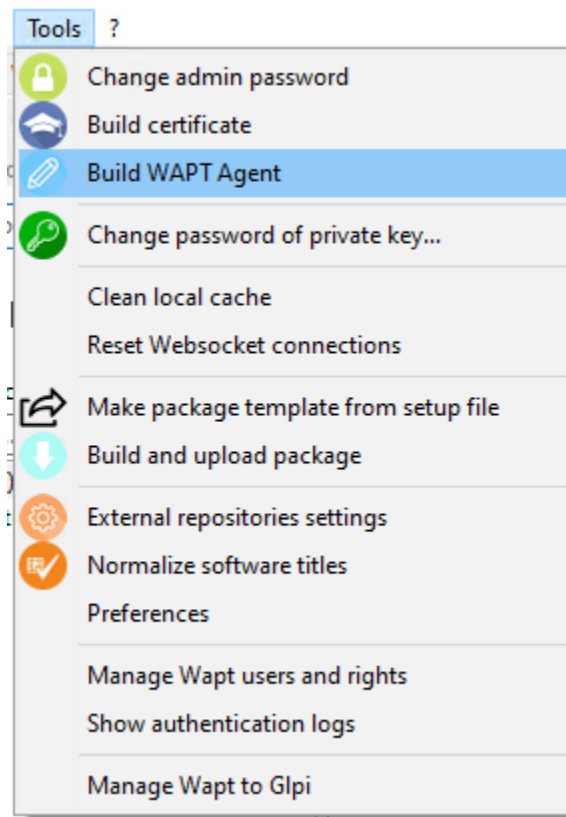


FIG. 1 – Générer l'agent WAPT depuis la console

## 16.1 Choix du mode d'identification unique des agents WAPT

Dans WAPT, vous pouvez choisir le mode d'identification unique des agents WAPT.

Lorsqu'un agent WAPT s'enregistre, le serveur doit savoir s'il s'agit d'une nouvelle machine ou d'une machine qui a déjà été enregistrée.

Pour cela, le serveur WAPT examine l'UUID (Universal Unique Identifier) de l'inventaire.

WAPT propose 3 modes pour vous aider à distinguer les machines, à vous de choisir le mode qui vous convient le mieux.

**Attention :** Après avoir choisi un mode de fonctionnement, il est difficile d'en changer, réfléchissez bien !

### 16.1.1 Identification des agents WAPT par leur UUID BIOS (numéro de série)

Ce mode de fonctionnement permet d'identifier les machines de la console de manière physique.

Si vous remplacez un ordinateur et donnez au nouvel ordinateur le même nom que le précédent, vous aurez deux ordinateurs qui apparaîtront dans la console WAPT puisque vous aurez physiquement deux ordinateurs différents.

---

**Note :** Certains fournisseurs font un travail inadéquat et attribuent les mêmes UUID de BIOS à des lots entiers d'ordinateurs. Dans ce cas, WAPT ne verra qu'un seul ordinateur !!!

---

### 16.1.2 Identification de l'agent WAPT par le nom d'hôte

Ce mode de fonctionnement est similaire à celui d'Active Directory. Les machines sont identifiées par leur nom d'hôte.

---

**Note :** Ce mode ne fonctionne pas si plusieurs machines de votre parc portent le même nom. Nous savons tous que cela ne devrait pas arriver !

---

### 16.1.3 Identification des agents WAPT à l'aide d'un UUID généré de manière aléatoire

Ce mode de fonctionnement permet d'identifier les PC par leur installation WAPT. Chaque installation de WAPT génère un numéro aléatoire unique. Si vous désinstallez WAPT puis le réinstallez, vous verrez apparaître un nouveau périphérique dans votre console.

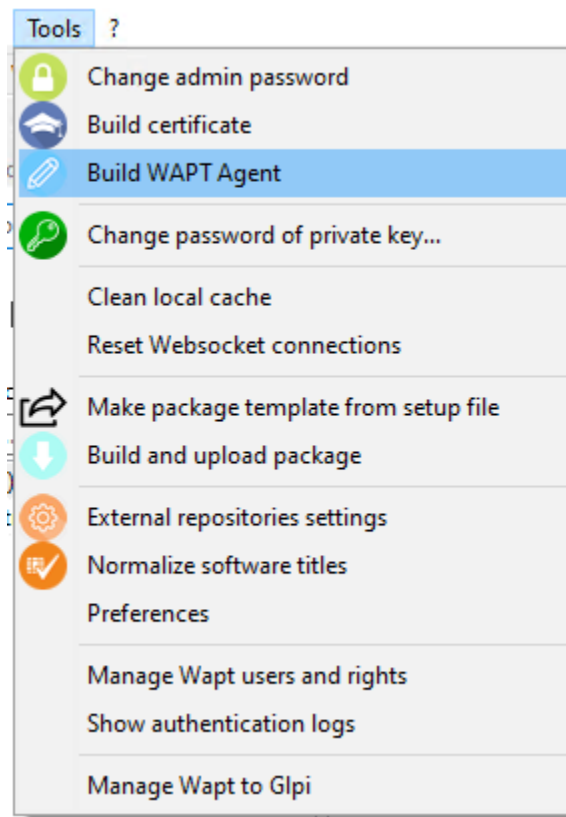


FIG. 2 – Générer l'agent WAPT depuis la console

## 16.2 Discovery

TABLEAU 1 – Informations sur l'agent WAPT

Valeur	Description	Requis
<i>Liste de certificats autorisés</i>	Dossier du certificat de confiance.	✓
<i>Inclure également les certificats utilisateurs</i>	Inclure le certificat WAPT local.	✗
<i>Adresse principale du dépôt WAPT</i>	Adresse du dépôt sur le serveur WAPT.	✓
<i>Adresse du serveur WAPT</i>	Adresse du dépôt sur le serveur WAPT.	✓
<i>Chemin vers les certificats d'autorité pour les serveurs https</i>	Chemin d'accès aux certificats utilisés pour la vérification HTTPS.	✗
<i>Organisation</i>	Nom de l'organisation permettant d'identifier l'origine des paquets WAPT.	✗
<i>Utiliser le nom FQDN de la machine comme identifiant UUID</i>	Si des FQDN sont utilisés pour <i>identifier les agents WAPT</i> .	✗
<i>Utiliser un UUID de machine aléatoire (pour les BIOS buggés)</i>	Si des UUID aléatoires sont utilisés pour <i>identifier les agents WAPT</i> .	✗

### Danger :

- La case à cocher **Utiliser kerberos pour l'enregistrement initial** doit être cochée **UNIQUEMENT SI** vous avez suivi la documentation sur **Configurer l'authentification kerberos**.
- La case à cocher **Vérifier le certificat HTTPS du serveur WAPT** doit être cochée **UNIQUEMENT SI** vous avez suivi la documentation sur **Activer la vérification du certificat SSL / TLS**.

— Fournissez le mot de passe pour déverrouiller la clé privée.

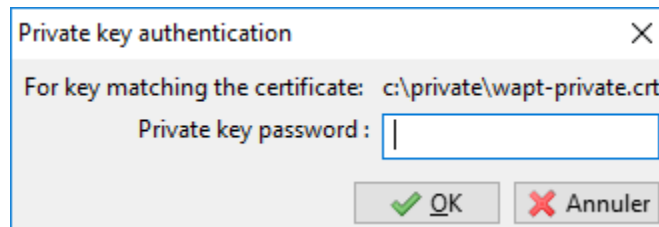


FIG. 3 – Fournissez le mot de passe pour déverrouiller la clé privée

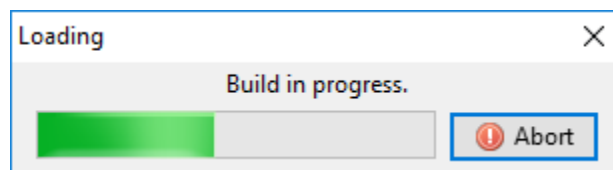


FIG. 4 – Progression de la construction de l'agent installateur WAPT

Une fois que le programme d'installation de l'agent WAPT a terminé sa construction, une boîte de dialogue de confirmation apparaît pour indiquer que le binaire **waptagent** a été téléchargé avec succès sur <https://srvwapt.mydomain.lan/wapt/>.

**Note :** Un avertissement s'affiche indiquant que la valeur de hachage du GPO doit être modifiée. Les GPO peuvent être utilisés pour

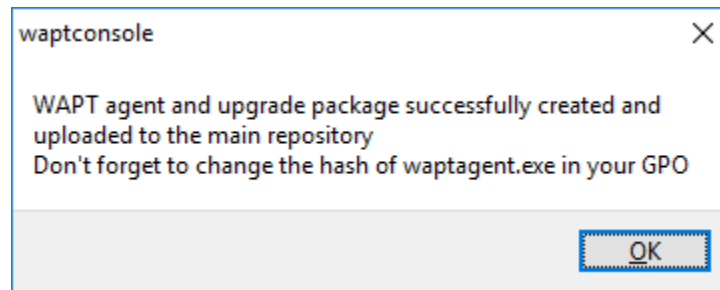


FIG. 5 – Confirmation de l'agent WAPT se chargeant du dépôt WAPT

déployer l'agent WAPT sur les ordinateurs de votre organisation.


---

**Danger :** Après avoir construit l'agent, installez le nouvel agent WAPT sur le poste exécutant la console de gestion WAPT.

## 16.3 Enterprise

— Remplissez les informations qui sont nécessaires pour l'installateur.

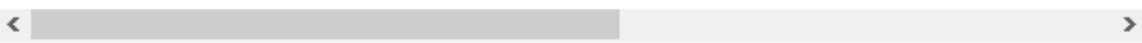
Create WAPT agent

Authorized packages certificates bundle :  

Include non CA too

Authorized packages certificates which will be bundled with the WAPT agent installer

▲	Certificate Name	Issuer	Valid until	Serial number	Fingerprint
0	test	test	2031-03-27 1...	476077107997...	91de981c7;


<  >

Main WAPT repository address :   Overwrite

WAPT server address :   Overwrite

Verify https server certificate

Use repository access rules

Path to https servers CA certificates bundle :  

Use Kerberos for initial registration

Organization :

Use computer FQDN for UUID

Use random host UUID (for buggy BIOS)

Always install these packages

Enable automatic install of packages based on AD groups

Allow remote reboot

Allow remote shutdown

Manage Windows updates with WAPT    Disable WAPT WUA    Don't set anything

WAPT WUA Windows updates

Allow all updates by default unless explicitly forbidden by rules

Scan / download scheduling :

Minimum delay before installation:  
(days after publish date)

Install pending Windows updates at shutdown

OK    Cancel

Fig. 6 – Complétez l'information sur votre Organisation

TABLEAU 2 – Informations sur l'agent WAPT

Valeur	Description	Requis
<i>Liste de certificats autorisés</i>	Dossier du certificat de confiance.	✓
<i>Inclure également les certificats utilisateurs</i>	Inclure le certificat WAPT local.	✗
<i>Adresse principale du dépôt WAPT</i>	Adresse du dépôt sur le serveur WAPT.	✓
<i>Adresse du serveur WAPT</i>	Adresse du dépôt sur le serveur WAPT.	✓
<i>Vérifier le certificat https serveur</i>	Si <i>L'authentification du client par certificat HTTPS</i> est activée sur le serveur WAPT.	✗
<i>Utiliser les règles d'accès aux dépôts</i>	Pour l'utilisation des règles de <i>réplication des dépôts distants</i> .	✗
<i>Chemin vers les certificats d'autorité pour les serveurs https</i>	Chemin d'accès aux certificats utilisés pour la vérification HTTPS.	✗
<i>Utiliser Kerberos pour l'enregistrement initial</i>	Si l'authentification <i>Kerberos</i> des agents WAPT est utilisée avec le serveur WAPT.	✗
<i>Organisation</i>	Nom de l'organisation permettant d'identifier l'origine des paquets WAPT.	✗
<i>Utiliser le nom FQDN de la machine comme identifiant UUID</i>	Si des FQDN sont utilisés pour <i>identifier les agents WAPT</i> .	✗
<i>Utiliser un UUID de machine aléatoire (pour les BIOS buggés)</i>	Si des UUID aléatoires sont utilisés pour <i>identifier les agents WAPT</i> .	✗
<i>Installer ces paquets sur les postes</i>	Installe automatiquement un <i>group</i> paquets lors de l'installation de l'agent WAPT.	✗
<i>Activer l'installation automatique de paquets basés sur les groupes AD</i>	Permet l'installation de <i>packages de profil</i> . <b>Cette fonctionnalité peut dégrader les performances de WAPT.</b>	✗
<i>Autoriser le redémarrage à distance</i>	Autoriser le redémarrage à distance depuis la console WAPT.	✗
<i>Autoriser l'extinction à distance</i>	Autoriser l'extinction à distance depuis la console WAPT.	✗
<i>Gérer les Windows Update avec WAPT`   Désactiver WAPT WUA   Ne rien changer</i>	Active ou désactive WAPT WUA.	✓
<i>Autoriser toutes les Mises à jour par défaut, sauf si celles explicitement exclus par les règles</i>	Autorise toutes les mises à jour de Windows si elles ne sont pas interdites par les paquets de règles WUA.	✗
<i>Planification du scan / téléchargement :</i>	Définit la périodicité de l'analyse de Windows Update.	✗
<i>Délai minimum avant l'installation (jours après la date de publication)</i>	Définit un délai d'installation différée avant la publication.	✗
<i>Installer les mises à jour Windows à l'arrêt</i>	Installe les mises à jour lorsque la machine s'éteint.	✗

**Indication :** Pour plus d'informations sur la section Windows update, consultez *cette article sur la configuration de WAPTWUA sur l'agent WAPT*

**Danger :**

- La case à cocher **Utiliser kerberos pour l'enregistrement initial** doit être cochée **UNIQUEMENT SI** vous avez suivi la documentation sur *Configurer l'authentification kerberos*.
- La case à cocher **Vérifier le certificat HTTPS du serveur WAPT** doit être cochée **SEULEMENT SI** vous avez suivi la documentation sur *Activer la vérification du certificat SSL / TLS*.

— Fournissez le mot de passe pour déverrouiller la clé privée.

Une fois que le programme d'installation de l'agent WAPT a terminé sa construction, une boîte de dialogue de confirmation apparaît



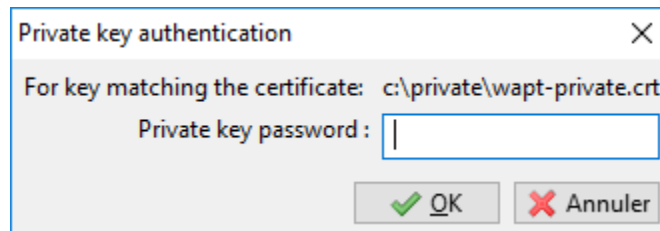


FIG. 7 – Fournissez le mot de passe pour déverrouiller la clé privée

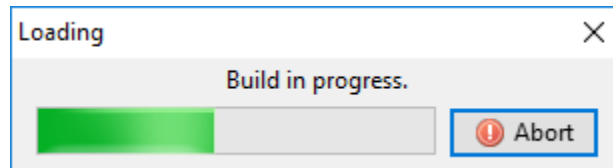


FIG. 8 – Progression de la construction de l'agent installateur WAPT

pour indiquer que le binaire **waptagent** a été téléchargé avec succès sur <https://srvwapt.mydomain.lan/wapt/>.

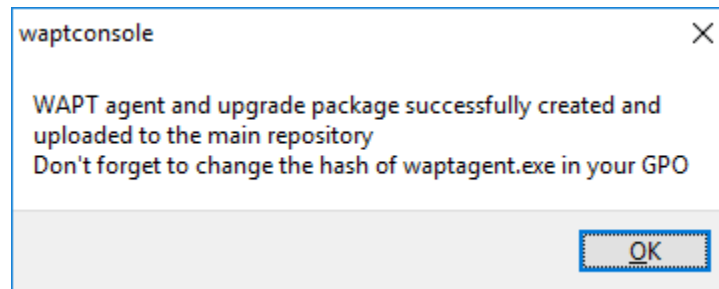


FIG. 9 – Confirmation de l'agent WAPT se chargeant du dépôt WAPT

**Note :** Un avertissement s'affiche indiquant que la valeur de hachage de la GPO doit être modifiée. Les GPO peuvent être utilisés pour déployer l'agent WAPT sur l'ordinateur de votre Organisation.

**Attention :** Après avoir construit l'agent sur votre ordinateur de gestion, quittez la console WAPT et *installer* le **nouvel agent WAPT** qui a été généré sur votre ordinateur de gestion WAPT.



---

## Configuration avancée de l'agent

---

Le fichier de configuration `wapt-get.ini` définit le comportement de l'agent WAPT.

TABLEAU 1 – Emplacement du `wapt-get.ini` par le système

Système	Localisation
Windows	C:\Program Files(x86)\wapt\wapt-get.ini
Linux	/opt/wapt/
Mac OS	/opt/wapt/

La section `[global]` est obligatoire.

```
[global]
```

Après l'installation standard, la configuration par défaut est la suivante :

```
[global]
waptupdate_task_period=120
wapt_server=https://srvwapt.mydomain.lan
repo_url=https://srvwapt.mydomain.lan/wapt/
use_hostpackages=1
```

Tous les paramètres ne sont pas disponibles lors de la génération de l'agent. Il est possible de faire des changements dans `wapt-get.ini` manuellement ou en déployant un paquet WAPT avec les nouveaux paramètres de configuration.

Un exemple de paquet est disponible dans le dépôt Tranquil IT :

<https://store.wapt.fr/store/tis-wapt-conf-policy>

```
# -*- coding: utf-8 -*-
from setuphelpers import *
```

(suite sur la page suivante)

```
uninstallkey = []

def install():

    print('Modify max_gpo_script_wait')
    inifile_writestring(WAPT.config_filename, 'global', 'max_gpo_script_wait', 180)

    print('Modify Preshutdowntimeout')
    inifile_writestring(WAPT.config_filename, 'global', 'pre_shutdown_timeout', 180)

    print('Disable Hyberboot')
    inifile_writestring(WAPT.config_filename, 'global', 'hiberboot_enabled', 0)

    print('Disable Notify User')
    inifile_writestring(WAPT.config_filename, 'global', 'notify_user', 0)

    print('Reload WAPT configuration')
    WAPT.reload_config_if_updated()
```

La définition de la fonction `inifile_writestring` est :

```
inifile_writestring(inifilename, section, key, value)
```

## 17.1 Description des sections disponibles

TABLEAU 2 – Description des sections disponibles pour l'agent WAPT

Section	Description
[global]	Options de l'agent global.
[wapt]	Options du dépôt principal.
[wapt-template]	Options du dépôt externe à distance.
[wapt-host]	Dépôt pour les paquets de l'hôte options.
[waptwua]	Options de l'agent WUA.
[repo-sync]	Options de dépôt multiple.







Toutes les sections sont détaillées ci-dessous.

## 17.2 Description des options disponibles par section

### 17.2.1 [global]

## Paramètres généraux

TABLEAU 3 – Description des options disponibles pour l’agent WAPT sur la section globale

Options / Valeur par défaut	Description	Exemple
 allow_remote_reboot = False	Permet de redémarrer les hôtes à distance à partir de la console WAPT (par défaut False).	allow_remote_reboot = True
 allow_remote_shutdown = False	Permet d’arrêter l’hôte à distance depuis la console WAPT (par défaut False).	allow_remote_reboot = True
check_certificates = False	Permet la vérification de la date et de la CRL du certificat du paquet.	check_certificates_validity = True
dbpath = WAPT root dir)\wapt\db\waptdb.sqlite	Chemin d’accès au fichier de la base de données locale.	dbpath = C:\Program Files(x86)\db\waptdb.sqlite
download_after_update = True	Définit si le téléchargement des paquets en attente doit être lancé après une mise à jour avec waptupdate_task_period.	download_after_update_with = False
 host_organizational_unit = None	Permet de forcer une unité organisationnelle sur l’agent WAPT (pratique pour attribuer un <i>unfaked OU</i> pour un PC hors-domaine). Assurez-vous qu’il respecte une casse cohérente (ne pas mélanger les « dc » s et les « DC » s, par exemple), que vous pouvez trouver dans la console (dans les champs DN/computer_ad_dn pour chaque hôte)	host_organizational_unit_dr = OU=TOTO,OU=TEST,DC=MYDC
 host_profiles = non défini	Permet de définir une liste de paquets WAPT que l’agent WAPT doit installer.	host_profiles = tis-firefox,tis-java
language = langue par défaut sur le client	Force la langue par défaut pour l’interface graphique (pas pour le filtrage des paquets)	language = fr
locales = locale par défaut sur le client	Permet de définir la liste des langues de l’agent WAPT pour pré-filtrer la liste des paquets visibles par l’agent WAPT (pour le filtrage des paquets). Le paramètre accepte plusieurs entrées (par exemple, locales=fr, en) ordonnées par préférence.	locales = en
log_to_windows_events = False	Permet d’écrire les journaux WAPT dans le journal des événements de Windows.	log_to_windows_events = True
loglevel = warning	Niveau de journalisation de l’agent WAPT. Les valeurs possibles sont : debug, info, warning, critical.	loglevel = critical
maturities = PROD	Liste des maturités de paquets qui peuvent être visualisées et installées par l’agent WAPT. La valeur par défaut est PROD. Seules les valeurs DEV, PREPROD et PROD sont utilisées par Tranquil IT, cependant toute valeur peut être utilisée pour s’adapter à vos processus internes.	maturities = PROD,PREPROD
repo_url = l’adresse de votre dépôt WAPT	Adresse du dépôt principal de WAPT.	repo_url = https://srvwapt.mydomain.lan/wapt
repositories = None	Liste des dépôts activés, séparés par une virgule. Chaque valeur définit une section du fichier wapt-get.ini. Plus d’info <i>ici</i> .	repositories = dépôt1, dépôt2
send_usage_report = True	Permet à la console WAPT d’envoyer des statistiques anonymes à Tranquil IT. Mettre à 0 pour désactiver la télémétrie.	send_usage_report = True
service_auth_type = system	Comment fonctionne l’authentification du self-service. Les valeurs possibles sont : system, waptserver-ldap ou waptagent-ldap	service_auth_type = waptserver-ldap
 uninstall_allowed = True	S’il est possible ou non pour l’utilisateur de désinstaller des applications via le self-service.	uninstall_allowed = False
 use_ad_groups = False	Pour l’utilisation de <i>paquet groupe</i> (par défaut False).	use_ad_groups = True
use_fqdn_as_uid	Permet d’utiliser le FQDN plutôt que l’UUID du BIOS comme identifiant unique	use_fqdn_as_uid

**Note :**

- S'il n'y a pas d'attribut `repo_url`, un dépôt dans la section `[wapt]` devra être explicitement défini. Il devra être activé en l'ajoutant à l'attribut `repositories`.
  - S'il n'y a pas d'attribut `wapt_server`, aucun serveur WAPT ne sera utilisé.
- 

**Paramètres du serveur**

Ces options définissent le comportement de l'agent WAPT lors de la connexion au serveur WAPT.

TABLEAU 4 – Description des options disponibles pour l'agent WAPT dans la section `[globale]` pour la configuration du serveur

Options / Valeur par défaut	Description	Exemple
<code>public_certs</code> = None	Dossier des certificats autorisés à vérifier la signature des paquets WAPT.	<code>public_certs_dir = C:\Program Files (x86)\wapt\ssl</code> (sous Windows) <code>public_certs_dir = /opt/wapt/ssl/</code> (sous Linux et MacOS)
<code>use_kerberos</code> = False	Utiliser l'authentification kerberos pour l'enregistrement initial sur le serveur WAPT (par défaut False).	<code>use_kerberos = True</code>
<code>verify_cert</code> = False	Voir la documentation sur l'activation de la <i>vérification des certificats HTTPS</i>	<code>verify_cert = True</code>
<code>wapt_server</code> = None	URL du serveur WAPT. Si cet attribut n'est pas présent, aucun serveur WAPT ne sera contacté.	<code>wapt_server = https://srvwapt.mydomain.lan</code>
<code>wapt_server_timeout</code> = 30	Délai de connexion HTTPS du serveur WAPT en secondes	<code>wapt_server_timeout = 10</code>

## paramètres waptexit

TABLEAU 5 – Description des options disponibles pour l’agent WAPT sur la section globale pour le waptexit

Options / Valeur par défaut	Description	Exemple
“al-low_cancel_upgrade” = True	Empêche les utilisateurs d’annuler les mises à jour des paquets à l’arrêt du poste. Si désactivé, les utilisateurs ne seront pas en capacité d’annuler les mises à jour des paquets à l’arrêt du poste. Si cette valeur n’est pas renseignée, elle sera par défaut a <b>10</b> .	“al-low_cancel_upgrade” = True
hiberboot_enabled = None	Désactive Hiberboot sur Windows 10 pour <b>waptexit</b> .	hiberboot_enabled = True
max_gpo_script_wait = None	Délai d’exécution des GPO à l’arrêt de l’ordinateur.	max_gpo_script_wait = 180
pre_shutdown_timeout = None	Délai d’attente pour les scripts à l’arrêt de l’ordinateur.	pre_shutdown_timeout = 180
“upgrade_only_if_not_processing” = False	Ne pas exécuter une mise à niveau du logiciel en cours d’exécution sur la machine (attribut <code>processing</code> du package)	“upgrade_only_if_not_processing” = True
upgrade_priorities = high	Réalise une mise a jour sur les paquets avec des propriétés spécifiques	upgrade_priorities = high
“waptexit_countdown” = 1	Temps avant le démarrage automatique des installations	“waptexit_countdown” = 25

## Paramètres d’authentification de Self-Service WAPT et Waptservice

TABLEAU 6 – Description des options disponibles pour l’agent WAPT dans la section globale pour le self-service WAPT et l’authentification Waptservice

Options / Valeur par défaut	Description	Exemple
ldap_auth_base_dn = None	Utile avec <i>waptagent-ldap</i> , définit le <i>dn de base</i> pour la requête LDAP.	ldap_auth_base_dn = dc=domain,dc=lan
ldap_auth_ssl_enabled = False	Utile avec <i>waptagent-ldap</i> , définit si la requête LDAP doit être chiffré.	ldap_auth_ssl_enabled = True
ldap_auth_server = None	Utile avec <i>waptagent-ldap</i> , définit le serveur LDAP à contacter.	ldap_auth_server = srvads.domain.lan
service_auth_type = system	Définit le système d’authentification du service WAPT, les valeurs disponibles sont <i>system</i> , <i>waptserver-ldap</i> , <i>waptagent-ldap</i> .	service_auth_type = waptagent-ldap
verify_cert_ldap = False	Utile avec <i>waptagent-ldap</i> , définit si le certificat doit être vérifié.	verify_cert_ldap = True
waptservice_admin_filter = False	Appliquer un filtrage d’affichage pour les <i>paquets self-service</i> pour les administrateurs locaux.	waptservice_admin_filter = True
waptservice_password = None	Mot de passe haché en sha256 lorsque <i>waptservice_user</i> est utilisé (la valeur <i>NOPASSWORD</i> désactive la nécessité d’un mot de passe).	waptservice_password = 5e884898da
waptservice_user = None	Force un utilisateur à s’authentifier sur le service WAPT.	waptservice_user = admin

## paramètres du wapttray

TABLEAU 7 – Description des options disponibles pour l’agent WAPT sur la section globale pour le wapttray

Options / Valeur par défaut	Description	Exemple
<code>notify_user = False</code>	Empêche wapttray d’envoyer des notifications (popup).	<code>notify_user = True</code>

## Paramètres du proxy

TABLEAU 8 – Description des options disponibles pour l’agent WAPT sur la section globale pour le proxy

Options / Valeur par défaut	Description	Exemple
<code>http_proxy = ""</code>	HTTP proxy address	<code>http_proxy = http://user:pwd@host_fqdn:port</code>
<code>use_http_proxy_for_repo = False</code>	Utilisez le proxy pour accéder aux dépôts.	<code>use_http_proxy_for_repo = True</code>
<code>use_http_proxy_for_server = False</code>	Utilisez un proxy pour accéder au serveur WAPT.	<code>use_http_proxy_for_server = True</code>

## Paramètres de création des paquets

TABLEAU 9 – Description des options disponibles pour l’agent WAPT dans la section globale pour la création de paquets WAPT

Options / Valeur par défaut	Description	Exemple
<code>default_package_prefix = tis</code>	Préfixe par défaut pour les paquets nouveaux ou importés.	<code>default_package_prefix = doc</code>
<code>default_sources_root = C:\waptdev (Windows) ou ~/waptdev (Linux)</code>	Répertoire pour le stockage des paquets en développement.	<code>default_sources_root = C:\waptdev</code>
<code>default_sources_suffix = wapt</code>	Suffixe par défaut pour les paquets nouveaux ou importés.	<code>default_sources_suffix = doc</code>
<code>personal_certificate_path = ""</code>	Chemin d’accès à la clé privée de l’administrateur.	<code>personal_certificate_path = None</code>

## 17.2.2 [wapt-wua]

Reportez-vous à *configurer WAPTWUA sur l’agent WAPT*.



### 17.2.3 Paramètres des dépôts supplémentaires

Liste complète des dépôts utilisables sur `wapt-get.ini`, une autre [section] peut être ajoutée.

---

**Note :** Les dépôts actifs sont listés dans l'attribut « repositories » de la section [global].

---

**Attention :** Ce paramètre peut être configuré à la fois dans la configuration de l'agent WAPT et dans le fichier de configuration de la console WAPT `C:\Users\%username%\AppData\Local\waptconsole\waptconsole.ini`.

Pour des informations sur la configuration de la console WAPT, veuillez vous référer à *cette élément de la documentation*.

#### [wapt]

---

**Indication :** Si cette section n'existe pas, les paramètres sont lus à partir de la section [global].

---

#### [wapt-templates]

Dépôts distants externes qui seront utilisés dans la console WAPT pour importer des nouveaux paquets ou leur mises à jour. Le dépôt Tranquil IT est défini par défaut.

#### [wapt-host]

Dépôt pour les paquets hôtes. Si cette section n'existe pas, les emplacements par défaut utilisés seront le dépôt principal.

Plus d'informations sur cette utilisation peuvent être trouvées dans *cette article sur le travail avec plusieurs dépôts publics ou privés*.



---

## Configuration de la console WAPT

---

**Indication :** la configuration de la console WAPT est stocké à 2 endroits :

- C:\Users\%username%\AppData\Local\waptconsole\waptconsole.ini.
- C:\Users\%username%\AppData\Roaming\waptconsole\waptconsole.ini.

Ces fichiers sont générés automatiquement lors du premier lancement de **waptconsole** et sont générés à partir du fichier `wapt-get.ini` configuré sur le poste de travail de l'Administrateur ;

---

### 18.1 Description des sections disponibles

TABLEAU 1 – Description des sections disponibles pour l'agent WAPT

Section	Description
[global]	options globales de la console
[sections]	options du dépôt externe. [wapt-template] est le <i>dépôt Tranquil IT</i> par défaut
[waptwua]	Options WUA

Toutes les sections sont détaillées ci-dessous.

Les autres sections présentes dans C:\Users\%username%\AppData\Roaming\waptconsole\waptconsole.ini ne sont pas modifiables manuellement, elles ne sont donc pas détaillées.

**Attention :** Pour les paramètres présents à la fois dans `wapt-get.ini` et `waptconsole.ini`, les valeurs sont définies dans `wapt-get.ini` et copiées dans `waptconsole.ini`. Ne modifiez pas manuellement ces paramètres.

## 18.2 Description des options disponibles par section


### 18.2.1 [global]

Plusieurs options sont disponibles dans la section [global] du fichier `waptconsole.ini`.

TABLEAU 2: Description des options disponibles sur AppData\Local

Options / Valeur par défaut	Description	Exemple
<code>advanced_mode = False</code>		Lance la console en mode débogage.
<code>allow_remote_reboot = False</code>		Permet <i>de redémarrer les hôtes à distance</i> à partir de la console WAPT.
<code>allow_remote_shutdown = False</code>		Permet <i>d'arrêter l'hôte à distance</i> depuis la console WAPT.
<code>client_certificate = None</code>		Si le dépôt distant utilise l'authentification SSL côté client
<code>client_private_key = None</code>		Si le dépôt distant utilise l'authentification SSL côté client
<code>check_certificates_validity = False</code>		Force la vérification de la date et de la CRL du certificat du dépôt
<code>default_maturity = ""</code>		Maturité de téléchargement par défaut pour les paquets WAPT
<code>default_package_prefix = tis</code>		Préfixe utilisé pour nommer les paquets WAPT.
<code>default_sources_root = C:\waptdev (Windows) ou ~/waptdev (Linux)</code>		Dossier de développement du paquet de base WAPT.
<code>grid_hosts_plugins = W10=</code>		<i>Plugins externes</i> pour la console WAPT. La valeur par défaut est W10=
<code>host_profiles = None</code>		Permet de définir une liste de paquets WAPT que l'agent WAPT peut installer
<code>hiberboot_enabled = False</code>		Désactive Hiberboot sur Windows 10 pour <b>waptexit</b>
<code>http_proxy = None</code>		Adresse du serveur proxy dans la console WAPT.
<code>last_usage_report = ""</code>		Date de la dernière utilisation de la console WAPT.
<code>lastwaptserveruser = ""</code>		Dernier utilisateur connecté sur cette console WAPT.
<code>max_gpo_script_wait = 180</code>		Délai d'exécution des GPO à l'arrêt de l'ordinateur.
<code>personal_certificate_path = ""</code>		Chemin d'accès au certificat associé à la clé privée de l'administrateur
<code>pre_shutdown_timeout = 180</code>		Délai d'attente pour les scripts à l'arrêt de l'ordinateur.
<code>repo_url = l'adresse de votre dépôt WAPT</code>		Adresse du dépôt principal de WAPT.
<code>send_usage_report = True</code>		Permet à la console WAPT d'envoyer des statistiques anonymes
<code>sign_digests = sha256</code>		Liste des algorithmes de signature autorisés pour les paquets
<code>use_ad_groups = False</code>		Pour utiliser les <i>paquets unit</i> .
<code>use_fqdn_as_uuid = False</code>		Permet d'utiliser le FQDN plutôt que l'UUID du BIOS comme identifiant
<code>use_kerberos = False</code>		Utilisez l'authentification kerberos pour l'enregistrement initial
<code>use_hostpackages = False</code>		Utilisation des <i>paquets host</i> .
<code>use_http_proxy_for_repo = False</code>		Utilisez un proxy pour vous connecter au dépôt principal de WAPT
<code>use_http_proxy_for_server = False</code>		Utilisez un proxy pour vous connecter au serveur WAPT à partir de la console
<code>use_repo_rules = False</code>		Pour les <i>dépôts secondaires</i> .
<code>verify_cert = False</code>		Pour la <i>vérification des certificats SSL / TLS</i> .
<code>wapt_server = ""</code>		Adresse du serveur WAPT.

TABLEAU 3 – Description des options disponibles sur AppData\Roaming

Options / Valeur par défaut	Description	Exemple
advanced_mode = False	Lance la console en mode débogage.	advanced_mode = True
enable_external_tools = False	Affiche les actions qui appellent des applications externes (RDP, outils Windows etc...).	enable_external_tools = True
enable_management_features = False	Affiche le bouton pour créer des certificats auto-signés ou pour créer l'installateur de l'agent WAPT.	enable_management_features = True
hide_unavailable_actions = False	Masque les actions qui ne sont pas disponibles pour l'agent WAPT	hide_unavailable_actions = True
HostsLimit = 2000	Limite des hôtes affichés dans la console WAPT.	HostsLimit = 300
language = langue par défaut sur le client	Forcer la langue par défaut pour l'interface graphique (pas pour le filtrage des paquets)	language = fr
lastappinifilename = ""	Le fichier ini actuellement utilisé par la console.	lastappinifilename = C:\Users\%username%\AppData\Roaming\waptconsole\waptconsole.ini
show_host_audit_data_tab = False	Affiche l'onglet <i>Données d'audit</i> sur l'inventaire des machines.	show_host_audit_data_tab = True
 use_ad_groups = False	Pour l'utilisation de <i>paquet unit</i> (par défaut False).	use_ad_groups = True
use_fqdn_as_uuid = False	Vous permet d'utiliser le nom fqdn plutôt que le BIOS uuid comme identifiant unique de la machine dans wapt (par défaut False).	use_fqdn_as_uuid = True
waptconsole.version = ""	Version de la console	waptconsole.version = 2.0.0.9424
waptwua_enabled = False	Pour afficher l'onglet Mises à jour Windows sur la console	waptwua_enabled = True

## 18.2.2 [sections]

Vous pouvez ajouter plusieurs dépôts externes en ajoutant [sections] dans C:\Users\%username%\AppData\Local\waptconsole\waptconsole.ini.

**Attention :** Ce paramètre peut être configuré à la fois dans la configuration de l'agent WAPT et dans la configuration de la console WAPT C:\Users\%username%\AppData\Local\waptconsole\waptconsole.ini.

Pour des informations sur la configuration de l'agent WAPT, veuillez vous référer à *ce point*.

Voir les paramètres et configurations disponibles *ici*.



## Configuration du serveur WAPT

Le fichier de configuration du serveur WAPT sur les systèmes GNU/Linux et macOS se trouve dans `/opt/wapt/conf/waptserver.ini` ou dans `/opt/wapt/waptserver/waptserver.ini`.

Le fichier de configuration du serveur WAPT sur les systèmes Windows se trouve dans `C:\wapt\conf\waptserver.ini`.

**Attention : La modification de ces fichiers est réservée aux utilisateurs avancés !**

### 19.1 Section [option] de waptserver.ini

Plusieurs options peuvent être définies dans cette section.

[options]

TABLEAU 1: Paramètres disponibles pour la section [option] de waptserver.ini

Options / Valeur par défaut	Description	Exemple
<code>allow_unauthenticated_connect = None</code>		Définit si les connexions non authentifiées sont autorisées
<code>allow_unauthenticated_registration = False</code>		Permet l'enregistrement de nouveaux utilisateurs
<code>allow_unsigned_status_data = False</code>		Débogage uniquement
<code>application_root = ""</code>		Définit un chemin relatif au répertoire de base
<code>auto_create_ldap_users = True</code>		Relatif aux ACL utilisateur
<code>client_certificate_lifetime = 3650</code>		Définit la durée de vie des certificats clients
<code>cleanup_kbs = True (implicite)</code>		Nettoie automatiquement les fichiers temporaires
<code>clients_read_timeout = 5</code>		Définit le délai d'attente des clients
<code>clients_signing_certificate = None</code>		Définit le chemin du certificat de signature des clients

suite sur la page suivante

Tableau 1 – suite de la page précédente

Options / Valeur par défaut	Description	Exemple
clients_signing_crl_days = 30		Définit la durée de v
clients_signing_crl = None		Définit le chemin de
clients_signing_crl_url = None		Définit l'URL de la
clients_signing_key = None		Définit le chemin de
client_tasks_timeout = 5		Définit le délai max
db_connect_timeout = 3		Définit le délai max
db_host = None		Définit l'url du serv
db_max_connections = 90		Définit l'url du serv
db_name = wapt		Définit la base de d
db_password = None		Définit le mot de pa
db_port = 5432		Définit le port du se
db_stale_timeout = 300		Définit le délai du t
db_user = wapt		Définit l'utilisateur
 enable_store = False		Active le WAPT Sto
encrypt_host_packages = False		Chiffre le paquet m
htpasswd_path = None		Ajoute l'authentific
http_proxy = None		Définit le serveur p
known_certificates_folder = dossier par défaut de WAPT /ssl/ folder		Ajoute une CA supp
ldap_auth_base_dn = None		Définit le DN de ba
ldap_auth_server = None		Définit le serveur d'
ldap_auth_ssl_enabled = True		Définit l'authentific
loglevel = warning		Définit le niveau du
max_clients = 4096		Définit la connexion
min_password_length = 10		Définit la longueur
nginx_http = 80		Définit le port <b>HTT</b>
nginx_https = 443		Définit le port <b>HTT</b>
remote_repo_support = False		Active la fonctionna
remote_repo_websockets = True		Permet la communi
secret_key = None		Définit la chaîne alé
server_uuid = None		Définit le serveur W
signature_clockskew = 300		Définit la différence
token_lifetime = 12*60*60		Définit la durée de v
trusted_signers_certificates_folder = None		Définit le chemin d'
trusted_users_certificates_folder = None		Définit le chemin d'
use_kerberos = False		Permet à un agent V
use_ssl_client_auth = False		Active l'authentifica
wapt_admin_group_dn = []		DN LDAP du group
wapt_folder = /var/www/wapt ou /var/www/html/wapt ou WAPT root_dir/waptserver/repository/wapt		Définit le chemin de
wapt_huey_db = None		Définit le chemin d'
wapt_password = None		Définit le mot de pa
waptserver_port = 8080		Définit le port du se
wapt_user = admin		Définit le nom d'uti
waptwua_folder = dossier_wapt + "wua"		Définit l'emplacem
wol_port = 9		Définit la liste des p
wapt_bind_interface = 127.0.0.1		Définit comment éc



## 19.2 Configuration de Nginx

La configuration par défaut de Nginx est la suivante :

```
server {
    listen          80;
    listen          443 ssl;
    server_name     _;
    ssl_certificate "/opt/wapt/waptserver/ssl/cert.pem";
    ssl_certificate_key "/opt/wapt/waptserver/ssl/key.pem";
    ssl_protocols  TLSv1.2;
    ssl_dhparam    /etc/ssl/certs/dhparam.pem;
    ssl_prefer_server_ciphers on;
    ssl_ciphers    'EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH';
    ssl_stapling   on;
    ssl_stapling_verify on;
    ssl_session_cache none;
    ssl_session_tickets off;
    index index.html;

    location ~ ^/wapt.* {
        proxy_set_header Cache-Control "store, no-cache, must-revalidate, post-check=0, pre-check=0";
        proxy_set_header Pragma "no-cache";
        proxy_set_header Expires "Sun, 19 Nov 1978 05:00:00 GMT";
        root "/var/www";
    }

    location / {
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location ~ ^/(api/v3/upload_packages|api/v3/upload_hosts/|upload_waptsetup) {
        proxy_pass http://127.0.0.1:8080;
        client_max_body_size 4096m;
        client_body_timeout 1800;
    }

    location /wapt-host/Packages {
        return 403;
    }

    location /wapt-host/add_host_kerberos {
        return 403;
    }

    location / {
        proxy_pass http://127.0.0.1:8080;
    }
}
```

(suite sur la page suivante)

```
location /socket.io {
    proxy_http_version 1.1;
    proxy_buffering off;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "Upgrade";
    proxy_pass http://127.0.0.1:8080/socket.io;
}
}
```

## 19.3 Configuration du serveur WAPT pour les grands déploiements

Les paramètres par défaut du système d'exploitation, de Nginx et de Postgresql sont adaptés pour environ 400 agents WAPT. Si vous avez plus de 400 clients, il est nécessaire de modifier quelques paramètres au niveau du système ainsi que la base de données PostgreSQL, le serveur web Nginx et le serveur python WAPT Server.

Dans le futur, le script **postconf.sh** pourra prendre en charge cette configuration en fonction du nombre d'ordinateurs clients attendus.

Avec les paramètres suivants, un serveur WAPT devrait pouvoir fonctionner avec environ 5000 clients actifs simultanés. Vous pouvez avoir plus de clients dans la base de données s'ils ne fonctionnent pas tous en même temps. Si vous avez plus de 5000 clients, il est recommandé d'avoir plus d'un serveur WAPT.

La limite du nombre de clients finaux est due à un goulot d'étranglement dans le code python et le backend PostgreSQL. Les performances de WAPT s'améliorent avec le temps et, à l'avenir, le serveur WAPT pourrait supporter une large base sur un seul serveur. Cependant, la partie Nginx s'adapte très bien et peut tirer pleinement parti d'une connexion 10Gbps pour les déploiements de paquets à forte charge.

---

**Note : Les paramètres à modifier ci-dessous sont liés entre eux et doivent être modifiés globalement et non individuellement.**

---

### 19.3.1 Configuration de Nginx

Dans le fichier `/etc/nginx/nginx.conf` (pour Windows `C:\wapt\waptserver\nginx\conf\nginx.conf`), modifiez le paramètre `worker_connections`. La valeur doit être environ 2,5 fois le nombre de clients WAPT (n connexions pour les websockets et n connexions pour les téléchargements de paquets et les envois d'inventaire + une certaine marge).

```
events {
    worker_connections 4096;
}
```

Mettez ensuite à niveau le nombre de `filedescriptors` dans le fichier `/etc/nginx/nginx.conf` (pour Windows `C:\wapt\waptserver\nginx\conf\nginx.conf`) :

```
worker_rlimit_nofile 32768;
```

En fonction du partitionnement de votre serveur WAPT, vous devrez peut-être faire attention au répertoire de téléchargement de fichiers temporaires de Nginx. Nginx agit comme un proxy inverse pour le moteur Python de WAPTServer et fait une mise en cache des paquets téléchargés lors du téléchargement d'un nouveau paquet depuis la console.

Les paquets sont stockés dans le répertoire `/var/lib/nginx/proxy`. Vous devez vous assurer que la partition qui héberge ce répertoire est suffisamment grande. Vous pouvez modifier l'emplacement de ce répertoire en utilisant le paramètre de configuration suivant de Nginx.

```
$client_body_temp_path
```

### 19.3.2 Configuration du système Linux

Augmenter le nombre de *filedescriptors*. Le fichier d'unité du système demande une augmentation du nombre autorisé de *filedescriptors* (`LimitNOFILE=32768`). Nous devrions avoir la même chose pour Nginx. Il y a quelques limites à modifier.

Tout d'abord, nous modifions au niveau du système le nombre de *filedescriptors* autorisés pour Nginx et WAPT.

— Créer `/etc/security/limits.d/wapt.conf`.

```
cat > /etc/security/limits.d/wapt.conf <<EOF
wapt      hard   nofile   32768
wapt      soft   nofile   32768
www-data  hard   nofile   32768
www-data  soft   nofile   32768
EOF
```

Nginx sert de proxy inverse et établit un grand nombre de connexions. Chaque client WAPT maintient une connexion *websocket* en permanence afin de répondre aux actions du serveur WAPT.

Le noyau Linux a une protection contre le fait d'avoir trop de connexions TCP ouvertes en même temps et on peut obtenir le message *SYN flooding on port* dans le journal de Nginx. Afin d'éviter ces messages, il est nécessaire de modifier les deux paramètres suivants. Il doit être environ 1,5 fois le nombre de clients WAPT.

```
cat > /etc/sysctl.d/wapt.conf <<EOF
net.ipv4.tcp_max_syn_backlog=4096
net.core.somaxconn=4096
EOF

sysctl --system
```

### 19.3.3 Configuration de la base de données PostgreSQL

Un plus grand nombre de clients nécessite un plus grand nombre de connexions à la base de données PostgreSQL. Dans le fichier `postgresql.conf` file (file `/etc/postgresql/11/main/postgresql.conf` sur debian 10 par exemple ou pour Windows `C:\wapt\waptserver\pgsql9.6_data\postgresql.conf`), vous devez augmenter le paramètre suivant pour atteindre approximativement 1/4 du nombre d'agents WAPT actifs.

```
max_connections = 1000
```

Dans le fichier `/opt/wapt/conf/waptserver.ini` (pour Windows `C:\wapt\conf\waptserver.ini`, `db_max_connections` doit être égal au `max_connections` de PostgreSQL moins 10 (PostgreSQL a besoin de garder quelques connexions pour son ménage). Le paramètre `max_clients` devrait être fixé à environ 1,2 fois le nombre d'agents WAPT :

```
[options]
...
```

(suite sur la page suivante)

(suite de la page précédente)

```
max_clients = 4096  
db_max_connections = 990
```

---

## Configuration des dépôts WAPT

---

### 20.1 Localisation du dépôt sur le serveur

Système d'exploitation	Valeur
Debian / Ubuntu	/var/www/wapt/
Centos / RedHat	/var/www/html/wapt/
Windows	C:\wapt\waptserver\repository

### 20.2 Répliquer un dépôt

#### 20.2.1 Aperçu fonctionnel

---

**Indication :** La méthode expliquée ci-dessous ne concerne que la version Enterprise.

La méthode *Synthing*, est dépréciée et **non supportée**, mais peut être utilisée pour les versions Discovery de WAPT.

---

## Rôle de réplication de l'agent WAPT

La réplication du dépôt peut être activée en utilisant un agent WAPT installé sur une machine existante, une appliance dédiée ou une machine virtuelle.

Le rôle de réplication est déployé par le biais d'un paquet WAPT qui active **le serveur web Nginx** et configure la planification, les types de paquets, la synchronisation des paquets, et bien plus encore.

Cette fonctionnalité permet aux agents WAPT de trouver dynamiquement leur dépôt WAPT disponible le plus proche à partir d'une liste de règles stockées sur le serveur WAPT.

## Comportement de réplication

La réplication du dépôt dans WAPT est gérée nativement par les agents WAPT.

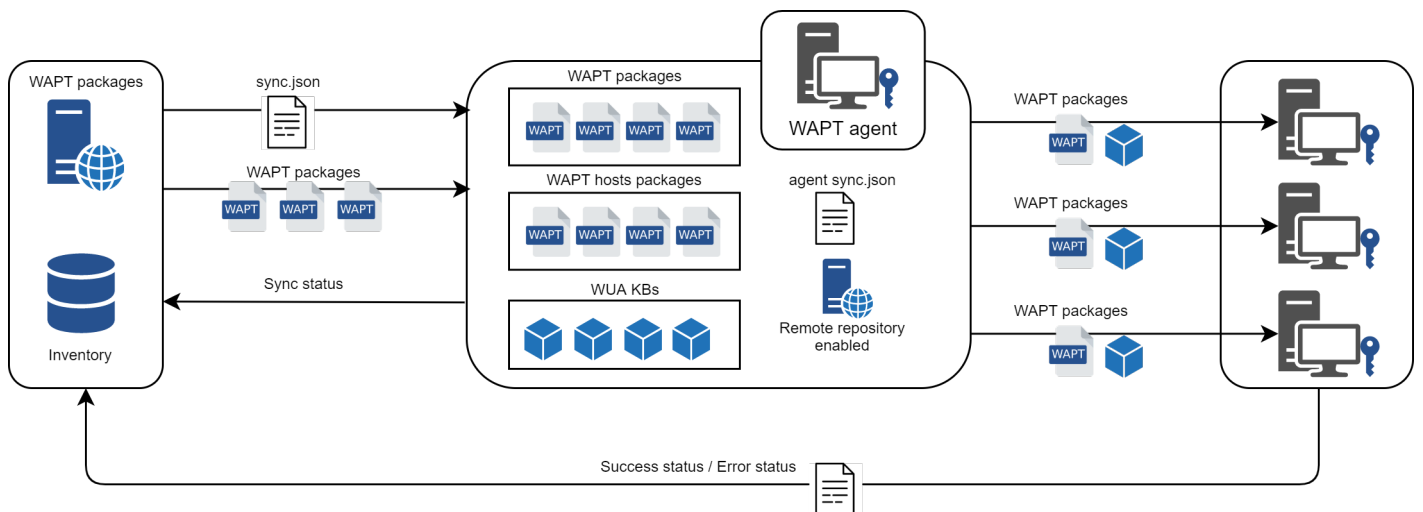
Il est basé sur un fichier `sync.json` qui indexe tous les fichiers présents dans ces dossiers :

- `wapt`;
- `waptwua`;
- `wapt-host`.

L'activation de la réplication a les effets suivants :

- Une fois que `enable_remote_repo` est activé sur un agent WAPT, il synchronisera les paquets localement dans le dossier `local_repo_path`.
- Il ajoute l'agent WAPT dans l'onglet *Dépôts secondaires* comme un dépôt distant, permettant de nouvelles actions telles que *Sync tous* ou *Créer l'index*.
- Par défaut, seul le dossier `wapt` est synchronisé, vous pouvez sélectionner le dossier à synchroniser en ajoutant des éléments dans les paramètres `remote_repo_dirs`.
- La période de synchronisation peut être configurée avec les paramètres `local_repo_time_for_sync_start` et `local_repo_time_for_sync_stop`.
- La bande passante allouée à la synchronisation peut être configurée avec `local_repo_limit_bandwidth`.

Tous les paramètres de la synchronisation du dépôt WAPT doivent être définis dans la section `[repo-sync]` du fichier de configuration `wapt-get.ini` de l'agent WAPT.



## 20.2.2 Configuration de l'agent WAPT

Pour activer la réplication sur un *Agent WAPT existant* (Linux / Windows) vous devez définir dans la section [repo-sync] dans le fichier de configuration `wapt-get.ini`.

TABLEAU 1 – Configuration de la réplication de l'agent WAPT

Options / Valeur par défaut	Définition	Exemple
<code>enable_remote_repo = False</code>	Permet au dépôt secondaire de se synchroniser avec le dépôt principal.	<code>enable_remote_repo = True</code>
<code>local_repo_path = Dossier de WAPT /repository</code>	Définit le chemin vers le répertoire racine du dépôt local pour les paquets WAPT.	<code>local_repo_path = /var/www/</code>
<code>local_repo_time_for_sync_start = None</code>	Définit l'heure de début de la synchronisation (HH:MM / format 24h)	<code>local_repo_time_for_sync_start = 22:30</code>
<code>local_repo_time_for_sync_end = None</code>	Définit l'heure d'arrêt de la synchronisation (HH:MM / format 24h)	<code>local_repo_time_for_sync_end = 05:30</code>
<code>local_repo_sync_task_period = None</code>	Définit la périodicité de la synchronisation (minutes)	<code>local_repo_sync_task_period = 25</code>
<code>local_repo_limit_bandwidth = None</code>	Définit la largeur de bande autorisée pour la synchronisation (Mbits/s)	<code>local_repo_limit_bandwidth = 2</code>
<code>remote_repo_dirs = wapt,waptwua</code>	Définit les dossiers à synchroniser	<code>remote_repo_dirs = wapt,waptwua,wapt-host</code>
<code>use_repo_rules = False</code>	Activer pour l'utilisation des <i>règles du dépôt</i>	<code>use_repo_rules = True</code>

**Avertissement :** Si vous modifiez manuellement le fichier `wapt-get.ini` sur le dépôt secondaire, vous devez redémarrer **wapt-service**.

**Note :** Un **Paquet WAPT** prêt à l'emploi est disponible sur le **store public Tranquill IT** pour permettre la réplication du dépôt sur des agents WAPT basés sur Windows ou Linux.

Ce paquet spécial :

- Installe et active le serveur web **Nginx** sur le dépôt secondaire.
- Configure l'environnement de l'hôte virtuel **Nginx**.
- Active la configuration du dépôt secondaire dans `wapt-get.ini`.

Il est possible de configurer automatiquement les dépôts avec vos propres valeurs en modifiant ce paquet.

Voici un exemple de `wapt-get.ini`.

```
[global]
...
use_repo_rules = True

[repo-sync]
enable_remote_repo = True
local_repo_path = D:\WAPT\
local_repo_time_for_sync_start = 20:30
local_repo_time_for_sync_end = 05:30
local_repo_sync_task_period = 25
```

(suite sur la page suivante)

```
local_repo_limit_bandwidth = 4
remote_repo_dirs = wapt,waptwua,wapt-host
```

### 20.2.3 Configuration du serveur WAPT

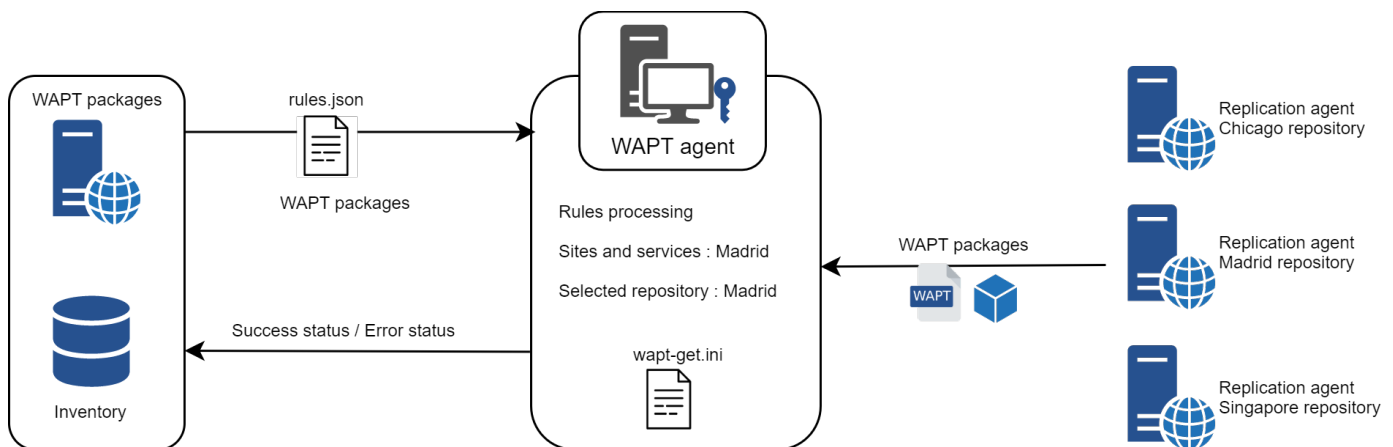
Par défaut, le serveur sait quels agents WAPT sont configurés en tant que dépôt secondaire et il les répertorie dans la console WAPT.

### 20.2.4 Règles du dépôt

Lorsqu'un agent WAPT a été configuré comme dépôt, il récupère automatiquement son fichier `rules.json` depuis le serveur WAPT.

Le fichier `rules.json` est un fichier `.json` signé qui contient une liste de règles triées à appliquer aux agents WAPT distants, afin qu'ils puissent se connecter aux dépôts les plus appropriés.

Si aucune règle ne correspond, l'agent WAPT se rabattra sur le paramètre `repo_url` du serveur WAPT défini dans le fichier de configuration `wapt-get.ini`.



### Agent WAPT

**Avertissement :** Si vous avez configuré des redirections GeoIP sur Nginx, vous devez les désactiver car elles peuvent entrer en conflit avec les règles du dépôt.

Pour activer les règles du dépôt de l'agent WAPT, vous devez activer ce paramètre dans la section `[global]` du fichier de configuration `wapt-get.ini` de l'agent WAPT.

Options / Valeur par défaut	Description	Exemple
<code>use_repo_rules = 0</code>	Pour l'utilisation de <i>réplication du dépôt</i> .	<code>use_repo_rules = True</code>

Voici un exemple de `wapt-get.ini`.



```
[global]
...
use_repo_rules = True
```

**Note :** Il est possible d'activer cette option lors de la *génération d'un agent WAPT*.

## Serveur WAPT

Sur le serveur WAPT, la fonctionnalité des dépôts secondaires est automatiquement activée.

Pour le contrôler, éditez `waptserver.ini` et lisez la valeur `remote_repo_support`.

Options / Valeur par défaut	Exemple de valeur	Définition
<code>remote_repo_support</code>	True	Permet l'utilisation du dépôt

## Console WAPT

Les règles du dépôt sont gérées à partir de la console WAPT et sont basés sur plusieurs paramètres :

TABLEAU 2 – Paramètres disponibles pour les règles du dépôt

Options	Exemple de valeur	Description
IP de l'agent	192.168.85.0/24	Règle basée sur le sous-réseau IP de l'agent.
Domaine	ad.mydomain.lan	Règle basée sur le nom de domaine Active Directory.
Nom d'hôte	desktop-04feb1	Règle basée sur le nom d'hôte de l'agent WAPT.
IP publique	256.89.299.22/32	Règle basée sur l'adresse IP publique (hôtes NATés).
Site	Paris-HQ	Règle basée sur les sites et services Active Directory.

## Ajout d'une règle

Dans *Dépôts*, cliquez sur le bouton *Ajouter une règle*. La fenêtre suivante apparaît.

FIG. 1 – Création d’une nouvelle règle de dépôt

TABLEAU 3 – Détail des valeurs

Options	Exemple de valeur	Description
<i>Name</i>	repo25	Définit le nom de la règle.
<i>Condition</i>	IP de l’agent	Définit la condition à remplir pour que la règle s’applique (voir ci-dessus).
<i>Valeur</i>	192.168.25.0/24	Définit la valeur lorsque la condition s’applique. Si la case « NON » est cochée, la valeur s’applique à l’inverse de la condition.
<i>URL du dépôt</i>	https://repo25.domain.lan	Définit la liste des dépôts secondaires disponibles. La liste inclut <code>http://download.windowsupdate.com/microsoftupdate/v6/wsuscan/</code> pour permettre le téléchargement direct des mises à jour de Windows par les dépôts secondaires afin de préserver la bande passante.
<i>Type de paquet</i>	WAPT	Définit quels <i>types de paquets</i> sont répliqués.
<i>Autre</i>	Pas de fallback	Voir ci-dessous

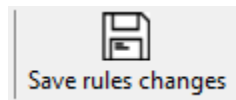
- L’option *Pas de fallback* empêchera de se replier sur le serveur WAPT principal et évitera une congestion indésirable du réseau si le dépôt secondaire devient temporairement indisponible.
- L’option *Proxy* devra être définie si le dépôt secondaire doit se connecter via un proxy.

Vous pouvez ensuite choisir parmi les différents paramètres ci-dessus et affecter des valeurs à un dépôt secondaire WAPT spécifique.

**Avertissement : Les règles sont appliquées de haut en bas. La première règle qui correspond aux conditions prévaut sur toutes les autres règles placées en dessous.**

Create new rule ✕  
 Name :   
 Condition :   
 Value :   NOT  
 Repository URL :   
 Package type :  WAPT  HOST  WUA  
 Other :  No fallback  Proxy  
 Proxy :

**Danger** : N'oubliez pas de sauvegarder vos règles de réplication.



## 20.3 Dépôts multiples

Comme pour les dépôts Debian, il est possible pour l'agent WAPT d'utiliser plusieurs dépôts pour la mise à jour des paquets. Les agents WAPT vérifieront tous les dépôts.

**Danger** : Si vous utilisez cette fonctionnalité, **SACHEZ CE QUE VOUS FAITES**.

Lorsque vous utilisez des dépôts avec différents signataires, les certificats publics du signataire supplémentaire doivent être ajoutés à C:\Program Files (x86)\wapt\ssl sous Windows ou /opt/wapt/ssl sous Linux et MacOS, par conséquent, vous **DEVEZ** faire confiance à leur travail et à leur signature.

Vous devez ensuite déployer l'agent WAPT avec les deux clés.

Veillez vous reporter à la documentation sur *la création de l'agent WAPT* pour ajouter d'autres certificats de confiance.

## 20.3.1 Configuration de l'agent WAPT

Ces paramètres sont modifiables dans le fichier `wapt-get.ini`.

### Description des paramètres disponibles

— [global]

paramètre *repositories* :

Le paramètre *repositories* permet de définir plusieurs options pour les dépôts de paquets, par exemple *wapt-templates* et *private*, où leurs paramètres sont définis dans une [section] supplémentaire du fichier.

```
repositories=wapt-templates,private
```

— [section]

paramètres des dépôts secondaires

```
[wapt-templates]
```

```
repo_url=https://store.wapt.fr/wapt
```

```
verify_cert = 1
```

```
[private]
```

```
repo_url=https://srvwapt.mydomain.lan/wapt
```

```
verify_cert = 0
```

Avec cette configuration, les clients verront maintenant les paquets disponibles sur le dépôt secondaire en plus du dépôt principal.

### Description des paramètres disponibles

TABLEAU 4 – Description des options disponibles pour l'utilisation de dépôts multiples

Options / Valeur par défaut	Description	Exemple
<code>http_proxy = None</code>	Définit l'adresse du proxy HTTP.	<code>http_proxy = http://user:pwd@host_fqdn:port</code>
<code>repo_url = None</code>	Définit l'adresse du dépôt WAPT principal.	<code>repo_url = https://srvwapt.mydomain.lan/wapt</code>
<code>timeout = None</code>	Définit le délai d'attente lors de la connexion à des dépôts distants. Valeur en millisecondes.	<code>timeout = 5000</code>
<code>use_http_proxy = False</code>	Définit si un proxy doit être défini pour accéder aux dépôts.	<code>use_http_proxy_for_repo = 1</code>
<code>verify_cert = None</code>	Définit si <i>Les certificats HTTPS du dépôt doivent être vérifiés</i> , et si c'est le cas définit le chemin vers le paquet de certificats.	<code>verify_cert = C:\Program Files (x86)\wapt\ssl\server\srvwapt.mydomain.lan.crt (sur Windows)</code>

**Note :** Les agents WAPT rechercheront les mises à jour de tous les dépôts.

```
wapt-get search
```

Plus d'informations sur *l'utilisation de WAPT avec l'interface de ligne de commande*.

**Indication :** Les paquets accessibles à partir de tous les dépôts définis seront également visibles à l'aide de l'interface web <http://127.0.0.1:8088> sur les appareils équipés de WAPT.

### 20.3.2 Configuration de la console WAPT

Après avoir configuré l'agent WAPT pour utiliser plusieurs dépôts, nous pouvons faire apparaître les dépôts dans la console WAPT.

Pour cela, modifier le fichier `%appdata%\local\waptconsole\waptconsole.ini`.

Exemple :

```
[wapt-template]
repo_url=https://wapt.tranquil.it/wapt
http_proxy=
verify_cert=1
public_certs_dir=
client_certificate=
client_private_key=
timeout=5

[private]
repo_url=https://srvwapt.mydomain.lan/wapt
http_proxy=
verify_cert=0
public_certs_dir=
client_certificate=
client_private_key=
timeout=5
```

TABLEAU 5 – Description des options disponibles pour les dépôts externes

Options / Valeur par défaut	Description	Exemple
<code>client_certificate</code> = None	Définit le dossier qui contient les certificats utilisés pour authentifier les paquets externes téléchargés.	<code>client_certificate = C:\Program Files (x86)\wapt\ssl\server\srwapt.mydomain.lan.crt</code> (sur Windows)
<code>client_private_key</code> = None	Définit le dossier qui contient la clé privée.	<code>client_private_key = C:\Program Files (x86)\wapt\ssl\server\srwapt.mydomain.lan.pem</code> (sur Windows)
<code>http_proxy</code> = None	Définit l'adresse du proxy à utiliser pour accéder au dépôt externe référencé dans la [section].	<code>http_proxy = http://proxy.mydomain.lan:8080</code>
<code>public_certs_dir</code> =	Définit le dossier qui contient les certificats utilisés pour authentifier les paquets externes téléchargés.	<code>public_certs_dir = C:\private</code>
<code>repo_url</code> = None	Définit l'adresse du dépôt WAPT externe.	<code>repo_url = http://srwapt.mydomain.lan/wapt</code>
<code>timeout</code> = None	Définit le délai d'attente pour le dépôt externe référencé dans la « [section] ». Si elle est laissée vide, la connexion n'aura pas de délai d'attente.	<code>timeout = 2</code>
<code>verify_cert</code> = None	Pour la <i>vérification des certificats HTTPS</i> .	<code>verify_cert = 1</code>

---

## Renforcer la sécurité de votre installation WAPT

---

Par défaut, tous les paquets WAPT sont signés avec votre clé privée, ce qui offre déjà un haut niveau de sécurité. Cependant, vous pouvez améliorer davantage la sécurité de WAPT.

Pour sécuriser complètement votre installation WAPT, vous devez procéder comme suit :

- Activez l'enregistrement authentifié pour filtrer les personnes autorisées à enregistrer le périphérique auprès du serveur WAPT.
- Activez la vérification du certificat https sur les agents et la console pour vous assurer que les agents WAPT et la console WAPT se connectent au bon serveur WAPT.
- Configurez l'authentification Active Directory pour permettre l'accès à la console WAPT uniquement aux administrateurs WAPT autorisés.
- Activez l'authentification par certificat côté client pour n'autoriser que les appareils authentifiés à accéder au serveur WAPT (Remarque : c'est particulièrement important si vous voulez exposer votre serveur WAPT à l'extérieur dans une DMZ (De-Militarized Zone)).
- Si vous utilisez la version **Enterprise** de WAPT et que vous exploitez une grande flotte avec plusieurs administrateurs, vous serez peut-être intéressé de savoir comment configurer et appliquer correctement les ACLs.

### 21.1 Configuration du pare-feu sur le serveur WAPT

La configuration du pare-feu du serveur WAPT est essentielle et devrait être la première étape pour obtenir une meilleure sécurité dans WAPT.

Comme WAPT vise à être sécurisé dès la conception, seul un ensemble minimal de *ports ouverts* est nécessaire sur le serveur WAPT par rapport aux autres solutions.

Vous trouverez dans la documentation suivante des conseils autour des configurations de pare-feu pour renforcer la sécurité du serveur WAPT.

### 21.1.1 Configuration du pare-feu pour le serveur WAPT sur Debian / Ubuntu

Par défaut sur Debian Linux, aucune règle de pare-feu ne s'applique.

— Désactivez **ufw** et installez **firewalld** à la place.

```
ufw disable
apt update
apt -y install firewalld
```

— Il suffit d'appliquer cette configuration **firewalld**.

```
systemctl start firewalld
systemctl enable firewalld
firewall-cmd --zone=public --add-port=80/tcp --permanent
firewall-cmd --zone=public --add-port=443/tcp --permanent
systemctl restart firewalld
```

### 21.1.2 Configuration du pare-feu pour le serveur WAPT sur RedHat / CentOS

— Il suffit d'appliquer cette configuration **firewalld**.

```
systemctl start firewalld
systemctl enable firewalld
firewall-cmd --zone=public --add-port=80/tcp --permanent
firewall-cmd --zone=public --add-port=443/tcp --permanent
systemctl restart firewalld
```

## 21.2 Configuration de l'authentification kerberos

---

#### Note :

- Sans l'authentification kerberos, vous devez soit faire confiance à l'enregistrement initial, soit saisir un mot de passe pour chaque poste de travail lors de l'enregistrement initial.
  - Pour plus d'informations, consultez la documentation sur *l'enregistrement d'une machine auprès du serveur WAPT et la signature des mises à jour d'inventaire*.
  - L'authentification kerberos sera utilisée uniquement lors de l'enregistrement du dispositif.
- 

### 21.2.1 Installation des composants kerberos et configuration du fichier krb5.conf

```
# Debian / Ubuntu
apt install krb5-user msktutil libnginx-mod-http-auth-spnego

# CentOS / RedHat
yum install krb5-workstation msktutil nginx-mod-http-auth-spnego
```



**Note :** L'enregistrement avec kerberos n'est pas disponible avec un serveur WAPT fonctionnant sous Windows.

Modifiez le fichier `/etc/krb5.conf` et remplacez tout le contenu par les 4 lignes suivantes en remplaçant `MYDOMAIN.LAN` par votre nom de domaine Active Directory (i.e. `<MYDOMAIN.LAN>`).

**Attention :** La valeur `default_realm` doit être écrit en MAJUSCULES!!!

```
[libdefaults]
default_realm = MYDOMAIN.LAN
dns_lookup_kdc = true
dns_lookup_realm=false
```

Récupérer un keytab de service. Utiliser les commandes **kinit** et **klist**. Vous pouvez utiliser un compte *Administrateur* ou tout autre compte ayant le droit délégué de joindre un ordinateur au domaine dans le conteneur de destination approprié (par défaut `CN=Computers`).

Dans la transcription shell ci-dessous, les commandes sont en noir et le texte renvoyé est commenté en gris clair :

```
sudo kinit administrator
## Password for administrator@MYDOMAIN.LAN:
## Warning: Your password will expire in 277 days on Mon. 17 sept. 2018 10:51:21 CEST
sudo klist
## Ticket cache: FILE:/tmp/krb5cc_0
## Default principal: administrator@MYDOMAIN.LAN
##
## Valid starting      Expires              Service principal
## 01/12/2017 16:49:31  02/12/2017 02:49:31  krbtgt/MYDOMAIN.LAN@MYDOMAIN.LAN
## renew until 02/12/2017 16:49:27
```

Si la demande d'authentification est réussie, vous pouvez alors créer votre Keytab HTTP avec la commande **msktutil**.

Veillez à modifier la chaîne `<DOMAIN_CONTROLLER>` avec le nom de votre contrôleur de domaine (par exemple : `sr-vads.mydomain.lan`).

```
sudo msktutil --server DOMAIN_CONTROLLER --precreate --host $(hostname) -b cn=computers --service_
→HTTP --description "host account for wapt server" --enctypes 24 -N
sudo msktutil --server DOMAIN_CONTROLLER --auto-update --keytab /etc/nginx/http-krb5.keytab --host
→$(hostname) -N
```

**Attention :** Assurez-vous d'avoir correctement configuré votre *nom d'hôte* de serveur WAPT avant d'exécuter ces commandes ; Afin de vérifier votre *nom d'hôte*, vous pouvez exécuter **echo \$(hostname)** et il doit retourner le nom qui sera utilisé par l'agent WAPT fonctionnant sur les stations de travail clientes.

— Appliquez les droits d'accès appropriés au fichier `http-krb5.keytab`.

```
#Debian / Ubuntu
sudo chmod 640 /etc/nginx/http-krb5.keytab
sudo chown root:www-data /etc/nginx/http-krb5.keytab
```

(suite sur la page suivante)

```
# CentOS / RedHat
sudo chown root:nginx /etc/nginx/http-krb5.keytab
sudo chmod 640 /etc/nginx/http-krb5.keytab
```

## 21.2.2 Post-configuration de kerberos pour le serveur WAPT

Vous pouvez maintenant utiliser le script de post-configuration pour configurer le serveur WAPT afin d'utiliser kerberos.

Le script de post-configuration va configurer **Nginx** et le serveur WAPT pour utiliser l'authentification kerberos.

---

**Indication :** Ce script de post-configuration doit être exécuté en tant que **root**.

---

```
/opt/wapt/waptserver/scripts/postconf.sh --force-https
```

L'authentification Kerberos sera maintenant configurée.

## 21.2.3 Cas particuliers d'utilisation

### Mon serveur WAPT n'a pas accès à un Active Directory en écriture

- Connectez-vous à votre Active Directory (pas un RODC).
- Créez un compte d'ordinateur *srvwapt*.
- Ajouter un SPN (Service Principal Name) sur le compte *srvwapt\$*.

```
setspn -A HTTP/srvwapt.mydomain.lan srvwapt
```

- Créer un keytab pour ce serveur WAPT.

```
ktpass -out C:\http-krb5.keytab -princ HTTP/srvwapt.mydomain.lan@MYDOMAIN.LAN rndpass -minpass 64 -
↪crypto all -pType KRB5_NT_PRINCIPAL /mapuser srvwapt$@MYDOMAIN.LAN
Reset SRVWAPT's password [y/n]? y
```

---

**Note :** Si l'adresse de votre serveur WAPT est différente de celle de votre domaine Active Directory, remplacez *HTTP/srvwapt.mydomain.lan@MYDOMAIN.LAN* par *HTTP/srvwapt.othername.com@MYDOMAIN.LAN*.

---

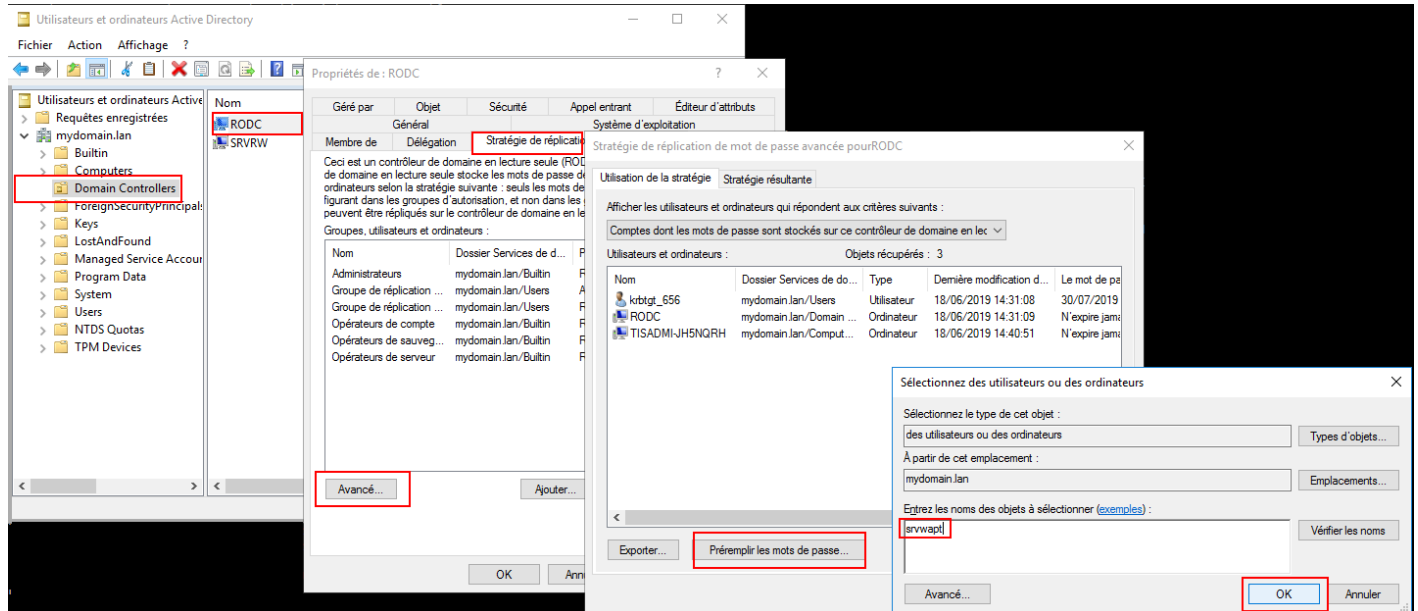
- Transférez ce fichier dans */etc/nginx/* (avec **winscp** par exemple).
- Appliquez les droits d'accès appropriés au fichier *http-krb5.keytab*.

```
# Debian / Ubuntu
sudo chmod 640 /etc/nginx/http-krb5.keytab
sudo chown root:www-data /etc/nginx/http-krb5.keytab

# CentOS / RedHat
sudo chown root:nginx /etc/nginx/http-krb5.keytab
sudo chmod 640 /etc/nginx/http-krb5.keytab
```

## L'agent WAPT n'a accès qu'à un contrôleur de domaine RODC

- Pour RODC (Read-Only Domain Controller), ajoutez le compte *srvwapt* au groupe de mots de passe autorisés pour la répliation.
- N'oubliez pas de précharger le mot de passe du serveur WAPT avec les différents serveurs RODC.



## Vous avez plusieurs domaines Active Directory, avec ou sans relations

Si vous avez plusieurs domaines Active Directory, vous devez créer un keytab par domaine en suivant la procédure ci-dessus, ex :

- `http-krb5-domain1.local.keytab` ;
- `http-krb5-domain2.local.keytab` ;
- `http-krb5-domain3.local.keytab`.

Vous devrez ensuite fusionner tous ces keytabs en un unique keytab :

```
ktutil
read_kt http-krb5-domain1.local.keytab
read_kt http-krb5-domain2.local.keytab
read_kt http-krb5-domain3.local.keytab
write_kt http-krb5.keytab
```

## 21.2.4 Débugger les problèmes avec les kerberos

### Attention :

- L'adresse du serveur ne peut pas être une IP, Kerberos ne fonctionne bien qu'avec le DNS.
- Dans votre test, l'url utilisée doit être **exactement** la même adresse que celle indiquée dans `C:\Program Files (x86)\waptwapt-get.ini`.

### Avez-vous redémarré nginx correctement ?

```
systemctl restart nginx
```

### Vérifier les permissions du fichier http-krb5.keytab

```
[root@srvwapt.mydomain.lan]# ls -l /etc/nginx/http-krb5.keytab
-rw-r----- 1 root www-data 921 janv.  4 16:20 /etc/nginx/http-krb5.keytab
```

### Le mode kerberos est-il actif sur mon agent ?

Sur la machine Windows :

- Vérifiez dans votre C:\Program Files (x86)\wapt\wapt-get.ini que la valeur use\_kerberos est True.

```
[global]
use_kerberos=1
```

- Si vous modifiez cette valeur, n'oubliez pas de redémarrer le service WAPT.

```
net stop waptservice
net start waptservice
```

### Le mode Kerberos est-il actif sur mon serveur ?

Sur la machine linux :

- Vérifiez dans votre /opt/wapt/conf/waptserver.ini que la valeur use\_kerberos est True.

```
[options]
use_kerberos=1
```

- Vérifiez dans votre /etc/nginx/sites-enabled/wapt.conf que cette configuration est présente.

```
location /add_host_kerberos {
    auth_gss on;
    auth_gss_keytab /etc/nginx/http-krb5.keytab;
    proxy_pass http://127.0.0.1:8080;
}
```

- Si l'une des deux configurations n'est pas présente, redémarrez la post-configuration et activez kerberos.

## Vérification que le fichier keytab contient l'url correcte

```
[root@srvwapt.mydomaine.lan]# KRB5_KTNAME=/etc/nginx/http-krb5.keytab klist -k
Keytab name: FILE:/etc/nginx/http-krb5.keytab
KVNO Principal
-----
...
 3 HTTP/srvwapt.ad.mydomain.lan@AD.MYDOMAIN.LAN
...

```

## Essayer d'enregistrer l'hôte en utilisant un compte système

Pour passer à un compte système, vous devez utiliser l'outil **psexec** de Microsoft : psexec.

— Dans **cmd** en tant qu'administrateur.

```
C:\Users\xxxxxx\Downloads\PSTools\psexec.exe -accepteula -s -i cmd
```

— Dans la nouvelle fenêtre **cmd**, vérifiez que vous êtes identifié comme *System*.

```
C:\WINDOWS\system32>whoami
```

```
NT AUTHORITY\System
```

— Exécutez la commande *register*.

```
wapt-get register
```

## Tenter une authentification avec le keytab de votre serveur WAPT

— Sur la machine linux.

```
[root@srvwapt.ad.tranq ~]# ktutil
ktutil: read_kt /etc/nginx/http-krb5.keytab
ktutil: list
slot KVNO Principal
-----
 1  3          srvwapt$@AD.TRANQUIL.IT
 2  3          srvwapt$@AD.TRANQUIL.IT
 3  3          srvwapt$@AD.TRANQUIL.IT
 4  3          SRVWAPT$@AD.TRANQUIL.IT
 5  3          SRVWAPT$@AD.TRANQUIL.IT
 6  3          SRVWAPT$@AD.TRANQUIL.IT
 7  3          host/srvwapt@AD.TRANQUIL.IT
 8  3          host/srvwapt@AD.TRANQUIL.IT
 9  3          host/srvwapt@AD.TRANQUIL.IT
10  3 HTTP/srvwapt.ad.tranquil.it@AD.TRANQUIL.IT
11  3 HTTP/srvwapt.ad.tranquil.it@AD.TRANQUIL.IT
12  3 HTTP/srvwapt.ad.tranquil.it@AD.TRANQUIL.IT
ktutil: quit
```

(suite sur la page suivante)

```

[root@srvwapt.ad.tranq ~]# kinit -k -t /etc/nginx/http-krb5.keytab srvwapt\@$AD.TRANQUIL.IT
[root@srvwapt.ad.tranq ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: srvwapt\@$AD.TRANQUIL.IT

Valid starting      Expires            Service principal
05/02/2021 19:06:05 06/02/2021 05:06:05  krbtgt/AD.TRANQUIL.IT@AD.TRANQUIL.IT
renew until 06/02/2021 19:06:05

```

## Tentative d'authentification avec curl

— Sur la machine linux.

```

[root@srvwapt.ad.tranq ~]# kdestroy
[root@srvwapt.ad.tranq ~]# kinit sfonteneau
Password for sfonteneau@AD.TRANQUIL.IT:
[root@srvwapt.ad.tranq ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: sfonteneau@AD.TRANQUIL.IT

Valid starting      Expires            Service principal
05/02/2021 19:10:42 06/02/2021 05:10:42  krbtgt/AD.TRANQUIL.IT@AD.TRANQUIL.IT
renew until 06/02/2021 19:10:39

root@srvwapt.ad.tranq ~]# curl -v --negotiate -u : https://srvwapt.ad.tranquil.it/add_host_kerberos_
↪-k
* Expire in 0 ms for 6 (transfer 0x563dece09f90)
* Uses proxy env variable no_proxy == 'localhost,127.0.0.1/8,192.168.0.0/16,10.0.0.0/8,172.16.0.0/12,
↪ad.tranquil.it'
* Expire in 1 ms for 1 (transfer 0x563dece09f90)
...
* Expire in 0 ms for 1 (transfer 0x563dece09f90)
* Expire in 0 ms for 1 (transfer 0x563dece09f90)
* Trying 192.168.149.37...
* TCP_NODELAY set
* Expire in 200 ms for 4 (transfer 0x563dece09f90)
* Connected to srvwapt.ad.tranquil.it (192.168.149.37) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* successfully set certificate verify locations:
* CAfile: none
CApath: /etc/ssl/certs
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Request CERT (13):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Certificate (11):

```

(suite sur la page suivante)

(suite de la page précédente)

```

* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* ALPN, server accepted to use http/1.1 Server certificate:
*  subject: C=FR; ST=PDLL; L=Saint Sebastien sur Loire; O=Tranquil IT Systems; OU=TIS; CN=srvwapt.
↪ad.tranquil.it; name=PKI TIS; emailAddress=technique@tranquil.it
*  start date: Aug  3 12:48:03 2017 GMT
*  expire date: Aug  1 12:48:03 2027 GMT
*  issuer: C=FR; ST=PDLL; L=Saint Sebastien sur Loire; O=Tranquil IT Systems; OU=TIS; CN=Tranquil_
↪IT Systems CA; name=PKI TIS; emailAddress=technique@tranquil.it
*  SSL certificate verify ok.
> GET /add_host_kerberos HTTP/1.1
> Host: srvwapt.ad.tranquil.it
> User-Agent: curl/7.64.0
> Accept: */*
>
< HTTP/1.1 401 Unauthorized
< Server: nginx
< Date: Fri, 05 Feb 2021 18:08:38 GMT
< Content-Type: text/html
< Content-Length: 188
< Connection: keep-alive
< WWW-Authenticate: Negotiate
< WWW-Authenticate: Basic realm=""
<
* Ignoring the response-body
* Connection #0 to host srvwapt.ad.tranquil.it left intact
* Issue another request to this URL: 'https://srvwapt.ad.tranquil.it/add_host_kerberos'
* Uses proxy env variable no_proxy == 'localhost,127.0.0.1/8,192.168.0.0/16,10.0.0.0/8,172.16.0.0/12,
↪ad.tranquil.it'
* Found bundle for host srvwapt.ad.tranquil.it: 0x563dece07590 [can pipeline]
* Could pipeline, but not asked to!
* Re-using existing connection! (#0) with host srvwapt.ad.tranquil.it
* Connected to srvwapt.ad.tranquil.it (192.168.149.37) port 443 (#0)
* Expire in 0 ms for 6 (transfer 0x563dece09f90)
* Server auth using Negotiate with user ''
> GET /add_host_kerberos HTTP/1.1
> Host: srvwapt.ad.tranquil.it
> Authorization: Negotiate_
↪IIgagYGKwYBBQUCoIIGXjCCBlqgDTALBgkqhkiG9xIBAgKiggZHBIIIGQ2CCBj8GCSqGSIB3EgECAGeAboIGLjCCBIqgAwIBBaEDAgEOogcD
↪+lGapoxK1hCC62Mnye9zF4SzOEKBDLPD77KJp2xbW2271c5ZF/
↪22wGXn8n6Sw1xndf1brq1mSEUo0TzpFUfy1wNoRDaw7WUNhQK2nbTTeMrsiiPACuQtG82W0VrZJZ+4z1Gq3ZFTLYUr1C010S1T8pNRzCLFR
↪OBq9EQd+i/2Mfp8XWY46GRtezTk8Dya+SH3henhB+L7G4ew0c3MKxFRkv0nXQ65qPAXWyogbivI/
↪ReekU1anHLnGfDyfeBw2QUM8t2kEEcSBmNKfrQ1U/u1jnlRZJ1o067PiziZsh7w/
↪zGpe7uh0a8a4RKYu1LeJEU+CKrifulQWkuqdiwIBdq9ApoQqduCbNse9ihH1srL3RYh9XdQ4Unx51eo49nZQA+c2Aj4JvCaFbY/
↪jeRBw6SNYCrFGETN1mXytjLryVBtJlch7djBGUAYaH1HGNfEVt+VnCW4090oqgCOM++u6d7Ci5w494ZseNXnF7RBKr01aVEt023lgeGg1Nv
↪arutH6c3CzLb+xPMAOUtCIup0M43SR0DC7gJ/
↪xZ3BZyKHF6b3p3tAWiByat2XNMxfMbgjaiV7oLCNEIAga0Igtg5f0nlahTI9323vfIH8aLNNVYVJeFKNGX1ord0YpJ3RLDshNBnoDTPyCKn

```

(suite sur la page suivante)

(suite de la page précédente)

```

→QXiFQaezmRut+hfXtoVch8LHC00IaloDUk1eHlFbAqQ980aE3SZ8Fx4m8Nw3JgQ0E+zXPt8DJCnNY4YV4j3+9b1093XhsJRyP97qEFazUGF
→PzqFnxSHtikXjCnjtzfuHLEPMWn0HKFbL/hEmmAnfZ15JiBgfBi820Xv3rCui+GKT/ZsjUFsgR8tCUJ58/gXBu9J/
→gY1R46CvWTn103+2JHqyomm6k0XnAU+s2hX+n/QcKbIjt7ew/f/
→UuT0J1YV+bQ8MMTPqQbau4f5sVaembIB7hTVyttpfbBEqCOV39xZ/r8b9CmpmukShPgeJI0x353i7b09/
→mBkchFaeyOc45ja7Z4iJ3IHniWLaWyYLktH/1N9/dXas8/CoZK0UsKjm9+xlkFFSP18CFHijlLlIc1sTdMnAil/
→jqwQl1W2WRnBltr/yE56EDR/i8VkcMHT5XsiiMCHm4LldkmDnIo/+GgHTG+3Z78Pkq939rMati/
→gzd9geYM8aUuYwJpcb53YsjWvJD1gDEHEwS3K1MYxbyy9eiODCOCgvIeKmVPouNrugXs4TX6PJsQDtZusSWxmZY4820HxmJkNT11G5Zkkt
→b93cbD4oa0WlbpVibPwLZ+TTckeGAxo3eBicENHSK81EIJYMBfEWTTsjYPEPsl5BK7IFcqArfEWG6HQDw3b0fyAB1ZJb0zSbhyD/
→rKnRmtSke/
→eWIAjYaeHDX0qYMruJCuI2lYofHtFwMEKSB1jCB06ADAgESooHLBIHIDVUTDas6nA0obxBuM2bQiZ0ZUPhAVGMOtniuCmBXU/
→mFRASD029zDjfl0nzeFsPdC4UBERcc8Vh4r3YeZixUxzn5tXCW4oFypYi5kHADx6Zd4GkZcEzAhRF7JwSylerZiCF+fnSiI5wdDG56PMF
> User-Agent: curl/7.64.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx
< Date: Fri, 05 Feb 2021 18:08:38 GMT
< Content-Type: text/html; charset=utf-8
< Content-Length: 38
< Connection: keep-alive
< WWW-Authenticate: Negotiate
→oYG3MIG0oAMKAQChCwYJKoZIhvcSAQICooGfBIGcYIGZBqkqhkiG9xIBAgICAG+BiTCBhqADAgEfoQMCAQ+iejB4oAMCARKicQRvQoZWpMI
→x0oFJX6n4DnhPZxrq/RnjwkoTnik7R8MjKRuvYncBfTGBIHvTJktq6+j9pHqmBDH5D5L8A
< WWW-Authenticate: Basic realm=""
< Cache-Control: store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
<
* Closing connection 0
kerberos connection seems to be working

```

## Vérification de la réussite de l'obtention d'un ticket Kerberos

**Attention :** Exécutez toujours les commandes dans le compte système (voir le point précédent) !

```

klist purge
klist get http/srvwapt.ad.mydomain.lan

```

Vous devez obtenir (dans votre langue) :

```

C:\Windows\System32>klist get http/srvwapt.ad.mydomain.lan

LogonId est 0:0x13794d
Un ticket pour http/srvwapt.ad.mydomain.lan a été récupéré.

Tickets mis en cache : (2)

#0> Client : sfonteneau @ AD.MYDOMAIN.LAN

```

(suite sur la page suivante)



(suite de la page précédente)

```
Serveur : krbtgt/AD.MYDOMAIN.LAN @ AD.MYDOMAIN.LAN
Type de chiffrement KerbTicket : AES-256-CTS-HMAC-SHA1-96
Indicateurs de tickets 0x40e00000 -> forwardable renewable initial pre_authent
Heure de démarrage : 2/4/2021 15:51:07 (Local)
Heure de fin : 2/5/2021 1:51:07 (Local)
Heure de renouvellement : 2/11/2021 15:51:07 (Local)
Type de clé de session : AES-256-CTS-HMAC-SHA1-96
Indicateurs de cache : 0x1 -> PRIMARY
KDC appelé : srvads.AD.MYDOMAIN.LAN
```

```
#1> Client : sfonteneau @ AD.MYDOMAIN.LAN
```

```
Serveur : http/srvwapt.AD.MYDOMAIN.LAN @ AD.MYDOMAIN.LAN
Type de chiffrement KerbTicket : AES-256-CTS-HMAC-SHA1-96
Indicateurs de tickets 0x40a80000 -> forwardable renewable pre_authent 0x800000
Heure de démarrage : 2/4/2021 15:51:07 (Local)
Heure de fin : 2/5/2021 1:51:07 (Local)
Heure de renouvellement : 2/11/2021 15:51:07 (Local)
Type de clé de session : AES-256-CTS-HMAC-SHA1-96
Indicateurs de cache : 0
KDC appelé : srvads.AD.MYDOMAIN.LAN
```

Si cela ne fonctionne pas, vérifiez dans votre Active Directory que l'attribut `serviceprincipalname` sur le compte de l'ordinateur du serveur WAPT a cette valeur : `HTTP/srvwapt.mydomain.lan`.

### Vérifiez qu'il fonctionne avec Firefox

---

**Note :** Vous devez d'abord configurer firefox pour l'authentification kerberos.

---

- Tapez **about:config** dans la barre d'URL de votre Firefox.
- Editez `network.negotiate-auth.trusted-uris`, et ajoutez l'url du serveur wapt : `srvwapt.mydomain.lan`.
- Vous pouvez maintenant visiter l'url : [https://srvwapt.mydomain.lan/add\\_host\\_kerberos](https://srvwapt.mydomain.lan/add_host_kerberos).
- Si l'authentification ne fonctionne pas, le serveur renvoie un message d'erreur 403.

### En cas d'erreur lors d'un des contrôles précédents

- Supprimez le compte de la machine de l'Active Directory.
- Supprimez le fichier `/etc/nginx/http-krb5.keytab`.
- Redémarrez la machine avec laquelle vous effectuez le test et exécutez à nouveau le processus de création de la keytab.

---

**Note :**

- Il est important de redémarrer la machine pour purger les tickets kerberos précédemment obtenus par la machine.
  - Pour éviter le redémarrage, vous pouvez également exécuter la commande « `klist purge` » en tant que SYSTEM.
-

## 21.3 Activation de la vérification du certificat SSL / TLS

Lors de l'exécution du script de post-configuration du serveur WAPT, le script générera un certificat auto-signé afin d'activer les communications HTTPS.

L'agent WAPT vérifie le certificat du serveur HTTPS en fonction de la valeur `verify_cert` de la section `[global]` dans `C:\Program Files (x86)\wapt\wapt-get.ini`.

TABLEAU 1 – Options pour `verify_cert`

Options pour <code>verify_cert</code>	Fonctionnement de l'agent WAPT
<code>verify_cert = 0</code>	l'agent WAPT ne vérifiera pas le certificat HTTPS du serveur WAPT.
<code>verify_cert = 1</code>	l'agent WAPT vérifiera le certificat HTTPS du serveur WAPT à l'aide du paquet de certificats <code>C:\Program Files (x86)\wapt\lib\site-packages\certifi\cacert.pem</code>
<code>verify_cert = C:\Program Files (x86)\wapt\ssl\srvwapt.mydomain.lan.crt</code>	l'agent WAPT vérifiera le certificat HTTPS du serveur WAPT avec le groupe de certificats <code>C:\Program Files (x86)\wapt\ssl\srvwapt.mydomain.lan.crt</code>

**Indication :** Pour activer rapidement et facilement la vérification du certificat https, vous pouvez utiliser la méthode *Pinning*.

### 21.3.1 Épingler le certificat

L'*épinglage de certificat* consiste à vérifier le certificat SSL/ TLS à l'aide de la définition d'un paquet bien défini et restrictif.

**Indication :** Cette méthode est la plus simple lorsqu'on utilise un certificat auto-signé.

Pour cela, vous devez lancer les commandes suivantes dans le shell Windows **cmd.exe** (avec des privilèges élevés si UAC (User Account Control) est actif).

Si vous avez déjà un shell Windows **cmd.exe** ouvert, fermez-le et ouvrez un nouveau shell afin de prendre en compte les variables d'environnement mises à jour :

```
wapt-get enable-check-certificate
net stop waptservice
net start waptservice
```

Validez le certificat avec **wapt-get update**

Lorsque vous avez exécuté la commande **update**, assurez-vous que tout s'est bien passé, et en cas de doute, vérifiez *Problème lors du enable-check-certificate*.

**Attention :** Si `wapt-get enable-check-certificate` renvoie une erreur, supprimez le `.crt` de même nom sur `C:\Program Files (x86)\wapt\ssl\server`

#### Note :

- La commande **enable-check-certificate** télécharge le certificat `srvwapt.mydomain.lan.crt` dans le dossier `C:\Program Files (x86)\WAPT\ssl\server`.

- Il modifie ensuite le fichier `wapt-get.ini` pour spécifier la valeur `verify_cert = C:\Program Files (x86)\wapt\ssl\server\srwapt.mydomain.lan.crt`.
- L'agent WAPT va maintenant vérifier les certificats en utilisant le certificat épinglé.

**Attention :** Si vous utilisez la méthode d'*épinglage de certificat*, n'oubliez pas de sauvegarder le dossier `/opt/wapt/waptserver/ssl` sur votre serveur WAPT.

Le fichier devra être restauré sur votre serveur si vous migrez ou mettez à niveau votre serveur WAPT, si vous voulez que les agents WAPT puissent continuer à établir des connexions HTTPS de confiance.

### 21.3.2 Comment utiliser un certificat commercial ou des certificats fournis par votre organisation ?

Si la méthode d'épinglage ne vous convient pas, vous pouvez remplacer le certificat auto-signé généré lors de l'installation de **WAPT**. Remplacez l'ancien certificat par le nouveau dans le dossier `/opt/wapt/waptserver/ssl/` (linux) ou `c:\wapt\waptserver\ssl\` (windows).

**La nouvelle paire de clés doit être au format PEM encodé en Base64.**

#### Note : Cas particulier où votre certificat a été signé par une Autorité de Certification interne

Les certificats émis par une *Autorité de certification* interne doivent avoir la chaîne de certificats complète de l'*Autorité de certification*.

Vous pouvez ajouter manuellement la chaîne de certificats de l'autorité de certification au certificat qui sera utilisé par **Nginx**.

Exemple : `echo srwapt.mydomain.lan.crt ca.crt > cert.pem`

- Pour les serveurs linux, il est également nécessaire de réinitialiser les ACLs :

```
# Debian / Ubuntu
chown root:www-data /opt/wapt/waptserver/ssl/*.pem

# CentOS / RedHat
chown root:nginx /opt/wapt/waptserver/ssl/*.pem
```

- Redémarrez **Nginx** pour prendre en compte les nouveaux certificats.

```
#Centos:
#Debian:
systemctl restart nginx

#Windows:
net stop waptnginx
net start waptnginx
```

## Configuration de l'agent WAPT

Pour un certificat commercial, vous pouvez définir `verify_cert = 1` dans `wapt-get.ini`.

Pour un certificat émis par une autorité de certification interne, vous devez placer le certificat dans le dossier `C:\Program Files (x86)\wapt\ssl\server\ca.crt` et spécifier le chemin du certificat avec `verify_cert` dans le fichier `:wapt-get.ini` de l'agent WAPT.

Pour appliquer la nouvelle configuration à l'ensemble de votre flotte :

- Régénérer un agent WAPT avec les paramètres appropriés.
- Utilisation du [paquet WAPT](#) pour modifier `wapt-get.ini` et pousser le certificat.

### 21.3.3 Vérification du certificat dans la console WAPT

Lorsque la console WAPT démarre pour la première fois, elle lit le contenu du fichier `C:\Program Files (x86)WAPT\wapt-get.ini` et elle construit son fichier de configuration `C:\Users\admin\AppData\Local\waptconsole\waptconsole.ini`.

Ceci définit correctement l'attribut `verify_cert` pour la communication HTTPS entre la console WAPT et le serveur WAPT.

## 21.4 Configuration de l'authentification des utilisateurs par rapport à l'Active Directory

Par défaut, le serveur WAPT est configuré avec un seul compte *SuperAdmin* dont le mot de passe est défini lors de la post-configuration initiale.

**Sur les réseaux étendus et sécurisés, ce compte SuperAdmin ne doit pas être utilisé car il ne peut pas fournir la traçabilité nécessaire aux actions administratives effectuées sur le réseau.**

Il est donc nécessaire de configurer l'authentification par rapport à l'Active Directory pour les utilisateurs de la console WAPT ; cela permettra d'utiliser des comptes nommés pour les tâches.

---

#### Note :

- L'authentification Active Directory est utilisée pour authentifier l'accès à l'inventaire via la console WAPT.
  - Cependant, toutes les actions sur les appareils distants équipés du WAPT sont basées sur des signatures X.509, donc un *Administrateur* aura besoin à la fois d'une connexion Active Directory **ET** d'une clé privée dont le certificat est reconnu par les appareils distants pour gérer sa base installée d'appareils utilisant WAPT.
  - Seul le compte *SuperAdmin* et les membres du groupe de sécurité Active Directory **waptadmins** seront autorisés à télécharger des paquets sur le dépôt principal (mode d'authentification par login et mot de passe).
- 

### 21.4.1 Activation de l'authentification Active Directory

- Pour activer l'authentification du serveur WAPT sur Active Directory, configurez le fichier `waptserver.ini` comme suit.
- 

**Note :** Le fichier de configuration du serveur WAPT sur les systèmes GNU/ Linux et macOS se trouve dans `/opt/wapt/conf/waptserver.ini` ou dans `/opt/wapt/waptserver/waptserver.ini`.

Le fichier de configuration du serveur WAPT sur les systèmes Windows se trouve dans `C:\wapt\conf\waptserver.ini`.

---

```
#waptserver.ini

wapt_admin_group_dn=CN=waptadmins,OU=groupes,OU=tranquilit,DC=mydomain,DC=lan
ldap_auth_server=srvads.mydomain.lan
ldap_auth_base_dn=DC=mydomain,DC=lan
ldap_auth_ssl_enabled=False
```

TABLEAU 2 – Options d’authentification disponibles

Options / Valeur par défaut	Description	Exemple
wapt_admin_group_dn = []	DN LDAP du groupe d’utilisateurs Active Directory autorisé à se connecter à la console WAPT	wapt_admin_group_dn = CN=waptadmins,OU=groups,DC=ad,DC=mydomain,DC=lan
ldap_auth_server = None	Définit le serveur d’authentification LDAP	ldap_auth_server = srvads.mydomain.lan
ldap_auth_base_dn = None	Définit le DN de base de l’authentification LDAP	ldap_auth_base_dn = dc=domain,dc=lan
ldap_auth_ssl_enabled = True	Définit l’authentification SSL sur les connexions LDAP	ldap_auth_ssl_enabled = False

— Redémarrer le service **waptserver**.

**Avertissement :** Pour **Microsoft Active Directory**, Microsoft a annoncé que l’authentification *SimpleBind* sur MS-AD sans SSL/TLS sera bloquée par défaut à partir d’avril 2020. Si vous n’avez pas de certificat installé, vous devrez modifier une clé de registre pour que l’authentification fonctionne.

**Note :** Par défaut **Samba-AD** ne permet pas l’authentification *SimpleBind* sans SSL/TLS. Si vous ne disposez pas d’un certificat valide, vous devrez modifier le paramètre `ldap server require strong auth` dans `/etc/samba/smb.conf`. Pour plus d’informations, vous pouvez consulter la documentation de Tranquil IT sur <https://dev.tranquil.it/samba/en/index.html>.

## 21.4.2 Activer le Single Sign On (SSO) pour la console WAPT et le selfservice

**Avertissement :** Cette configuration n’est disponible que pour les serveurs sous Linux : CentOS, Debian ou Ubuntu.

Vous pouvez utiliser Kerberos pour vous authentifier sur la **console WAPT** et le **selfservice**. De cette façon, les utilisateurs n’ont plus besoin de saisir leur mot de passe.

**Indication :** Il n’est pas nécessaire d’utiliser Kerberos pour l’enregistrement des agents afin d’utiliser le SSO.

## Préparer le serveur pour Kerberos Single Sign On

**Attention :** Pour activer Kerberos sur le serveur WAPT avec l'option `use_kerberos = True`, veuillez vous référer à *la configuration de l'authentification Kerberos* au préalable.

Il existe 3 méthodes pour configurer votre serveur WAPT avec Kerberos et l'authentification LDAP.

Pour chacun d'entre eux, vous devrez modifier le fichier `waptserver.ini`.

1. **La première méthode** est la moins sécurisée .

Avec ces options, vous ne vérifiez pas la certification ldap et n'utilisez pas de port sécurisé pour requêter le serveur :

```
ldap_auth_ssl_enabled=False
verify_cert_ldap=False
```

En effet, `ldap_auth_ssl_enabled=False` n'essayera pas de requêter l'Active Directory avec le protocole LDAPS.

L'option `verify_cert_ldap=False` est définie si vous n'utilisez pas *SSL/TLS support*.

---

**Indication :** Si votre serveur Active Directory est un Samba-AD et que vous avez cette option dans le `waptserver.ini` :

```
ldap_auth_ssl_enabled=False
```

Le serveur Samba refusera alors la connexion.

Par défaut **Samba-AD** ne permet pas l'authentification *SimpleBind* sans SSL/TLS.

Si vous n'avez pas de certificat valide, vous devrez modifier le paramètre `ldap server require strong auth` dans `/etc/samba/smb.conf`.

Pour plus d'informations, vous pouvez consulter la documentation de Tranquil IT sur <https://dev.tranquil.it/samba/fr/index.html>.

---

2. **La deuxième façon** plus sûre mais pas parfaite.

Il s'agit d'activer l'authentification ssl mais sans vérification de la certification :

```
ldap_auth_ssl_enabled = True
verify_cert_ldap=False
```

Le serveur WAPT essaiera d'utiliser le protocole LDAPS mais sans vérification de certificat pour contacter Active Directory.

3. **La méthode recommandée** est la plus sûre.

Elle utilise ces options :

```
ldap_auth_ssl_enabled = True
verify_cert_ldap=True
```

Mais pour faire fonctionner cela, vous allez devoir *activer le support SSL/TLS*.

Enfin, vous devrez ajouter ces options :

```
ldap_account_service_login = wapt-ldap@ad.tranquil.it
ldap_account_service_password = PASSWORD
ldap_auth_server = srvads.mydomain.lan
ldap_auth_base_dn =dc=mydomain,dc=lan
use_kerberos = True
```

Puis redémarrez les services sur le serveur :

```
systemctl restart waptserver wapttasks
```

**Note :** Les options `ldap_account_service_login` et `ldap_account_service_password` nécessitent un compte utilisateur dans votre Active Directory.

Il n'est pas nécessaire d'avoir un compte avec des droits élevés, juste un compte d'utilisateur en tant que service afin d'obtenir les groupes des utilisateurs.

## Configuration de l'agent WAPT

Du côté du client, vous allez devoir vous assurer que ces 2 options sont définies dans `wapt-get.ini` :

```
service_auth_type=waptserver-ldap
use_kerberos =1
```

Il est possible de faire des changements dans `wapt-get.ini` manuellement ou en déployant un paquet WAPT avec les nouveaux paramètres de configuration.

Un [paquet d'exemple](#) est disponible dans le dépôt Tranquil IT.

Avec cette configuration, vous pouvez lancer votre console WAPT ou votre selfservice sans demander de mot de passe.

**Note :** La console WAPT continuera à vous demander un login/mot de passe : c'est tout à fait normal, de cette façon vous pouvez utiliser un autre utilisateur que votre utilisateur actuel dans votre session Windows.

Sinon, il vous suffit de mettre votre login et de cliquer sur OK.

## 21.4.3 Activez le support SSL/ TLS pour les connexions LDAP dans le Contrôleur de Domaine Active Directory

Par défaut, l'authentification sur Active Directory repose sur LDAP SSL (port 636 par défaut).

SSL/ TLS n'est pas activé par défaut sur Microsoft Active Directory tant qu'un certificat SSL n'a pas été configuré pour le contrôleur de domaine.

**Note :** Le serveur WAPT utilise les *paquets* d'autorité de certification du système d'exploitation (CentOS) pour valider la connexion SSL/ TLS à Active Directory.

Si le certificat Active Directory est auto-signé ou a été signé par une autorité de certification interne, vous devrez ajouter ces certificats au magasin de certificats.

Ajouter un *Autorité de Certification* dans le dossier `/etc/pki/ca-trust/source/anchors/` et mettez à jour le magasin des CA.

```
# Debian / Ubuntu
cp cainterne.crt /usr/local/share/ca-certificates/cainterne.crt
update-ca-certificates

# CentOS / RedHat
cp cainterne.crt /etc/pki/ca-trust/source/anchors/cainterne.crt
update-ca-trust

# Windows
certutil -addstore -f "ROOT" cainterne.crt
```

- Une fois que vous avez configuré LDAP SSL/ TLS sur votre Active Directory (veuillez vous référer à la documentation de Microsoft pour cela), vous pouvez activer le support de la sécurité SSL/TLS pour AD dans `waptserver.ini`.

```
ldap_auth_ssl_enabled = True
```

- Redémarrer le service **waptserver**.

## 21.5 Configuration de l'authentification par certificat côté client

Si votre entreprise a besoin d'un serveur WAPT ouvert sur Internet, il peut être sécurisé grâce à l'authentification par certificat côté client.

Cette configuration restreint la visibilité du serveur WAPT aux seuls clients enregistrés. Cela se fait en s'appuyant sur la clé privée de l'agent WAPT générée lors de l'enregistrement. Elle fonctionne comme suit :

- L'agent WAPT envoie un CSR (Certificate Signing Request) au serveur WAPT qui le signe et le renvoie à l'agent WAPT.
- Grâce au certificat signé, l'agent peut accéder aux parties protégées du serveur Web **Nginx**.

**Note :** Nous recommandons fortement d'activer l'enregistrement Kerberos ou par login/mot de passe dans la post-configuration du serveur WAPT.

**Avertissement :** Toutes les actions sont à mener sur le serveur WAPT

### 21.5.1 Activation de l'authentification des certificats côté client sur le serveur WAPT

**Avertissement :** Pour **Linux**, vérifiez si le lien symbolique dans `sites-enabled` existe :

```
cd /etc/nginx/sites-enabled/
find . -maxdepth 1 -type l -ls
```

Le résultat escompté devrait être :

```
269091      0 lrwxrwxrwx  1 root    root          36 juil. 22 15:51 ./wapt.conf -> /etc/nginx/
↳ sites-available/wapt.conf
```

Sinon, utilisez la commande suivante :



```
ln -s /etc/nginx/sites-available/wapt.conf ./wapt.conf
```

Pour activer l'authentification, vous devez ajouter ces paramètres dans *le fichier de configuration* du serveur WAPT dans la section option :

```
use_ssl_client_auth = True
```

Relancez le script post-configuration .

**Indication :** Attention, à la date du 2024-01-09, WAPT ne supporte pas les CRL, ce qui signifie que lorsque vous supprimez une machine dans la console WAPT, la machine aura toujours accès au dépôt WAPT.

**Avertissement :** WAPTDeploy ne peut pas utiliser le https pour récupérer l'agent WAPT, vous devrez ajouter cette section dans le fichier :

```
server {
    listen            80;
    listen            [::]:80;
    server_name      -;

    location ~ ^/(wapt/waptsetup-tis.exe|wapt/waptagent.exe|wapt/waptdeploy.exe)$ {
        add_header Cache-Control "store, no-cache, must-revalidate, post-check=0, pre-check=0";
        add_header Pragma "no-cache";
        root "/var/www";
    }

    return 301          https://$host$request_uri;
}
```

## 21.6 Génération de l'autorité de certification (CA)

Lors de l'installation de WAPT, il vous est demandé de *créer* une paire *.pem / .crt* en cochant les cases *Pour Signature de code* et *Pour usage en tant que CA*.

Cette paire *.pem / .crt* permettra de signer les paquets WAPT et les nouveaux certificats.

## 21.6.1 Generating a new certificate with the Certificate Authority

Construire une nouvelle paire `.pem / .crt`.

---

**Note :** Le nouveau certificat ne sera pas un certificat auto-signé ;

Ce nouveau certificat sera signé par le CA (la clé générée lors de la première installation de WAPT) ;

---

Vous devez ensuite remplir la *Clé privée de l'autorité* et le *Certificat de l'autorité*.

Lors de la génération de la nouvelle paire pem/ crt, vous avez la possibilité de choisir si le nouveau certificat sera de type **Pour Signature de code** ou non.

---

**Indication :** Pour rappel, un certificat *Pour Signature de code* est réservé aux personnes ayant le rôle *Administrateur* dans le contexte de WAPT et un simple certificat SSL sans l'attribut *Pour Signature de code* est réservé aux personnes ayant le rôle *Déployeur de paquet*.

L'*Administrateurs* sera autorisé à signer les paquets qui **CONTIENNENT** un fichier exécutable `setup.py` (c'est-à-dire les paquets *base*).

Les personnes ayant le rôle *Déployeur de paquet* seront autorisées à signer les paquets qui **NE CONTIENNENT PAS** le fichier exécutable `setup.py` (c'est-à-dire les paquets *host, unit* et *group*).

---

Les clés et les certificats qui ne sont pas **Signature de code** peuvent être distribués aux personnes chargées de déployer les paquets sur la base installée des appareils équipés de WAPT.

Une autre équipe disposant de certificats ayant l'attribut **Pour Signature de code** préparera les paquets WAPT contenant les applications qui devront être configurées conformément aux directives de sécurité de l'*Organisation* et aux personnalisations utilisateur souhaitées par celle-ci.

La génération d'une nouvelle paire `.pem / .crt` permettra également d'identifier formellement la personne qui a signé un paquet en recherchant l'attribut CN du certificat de paquet WAPT.

---

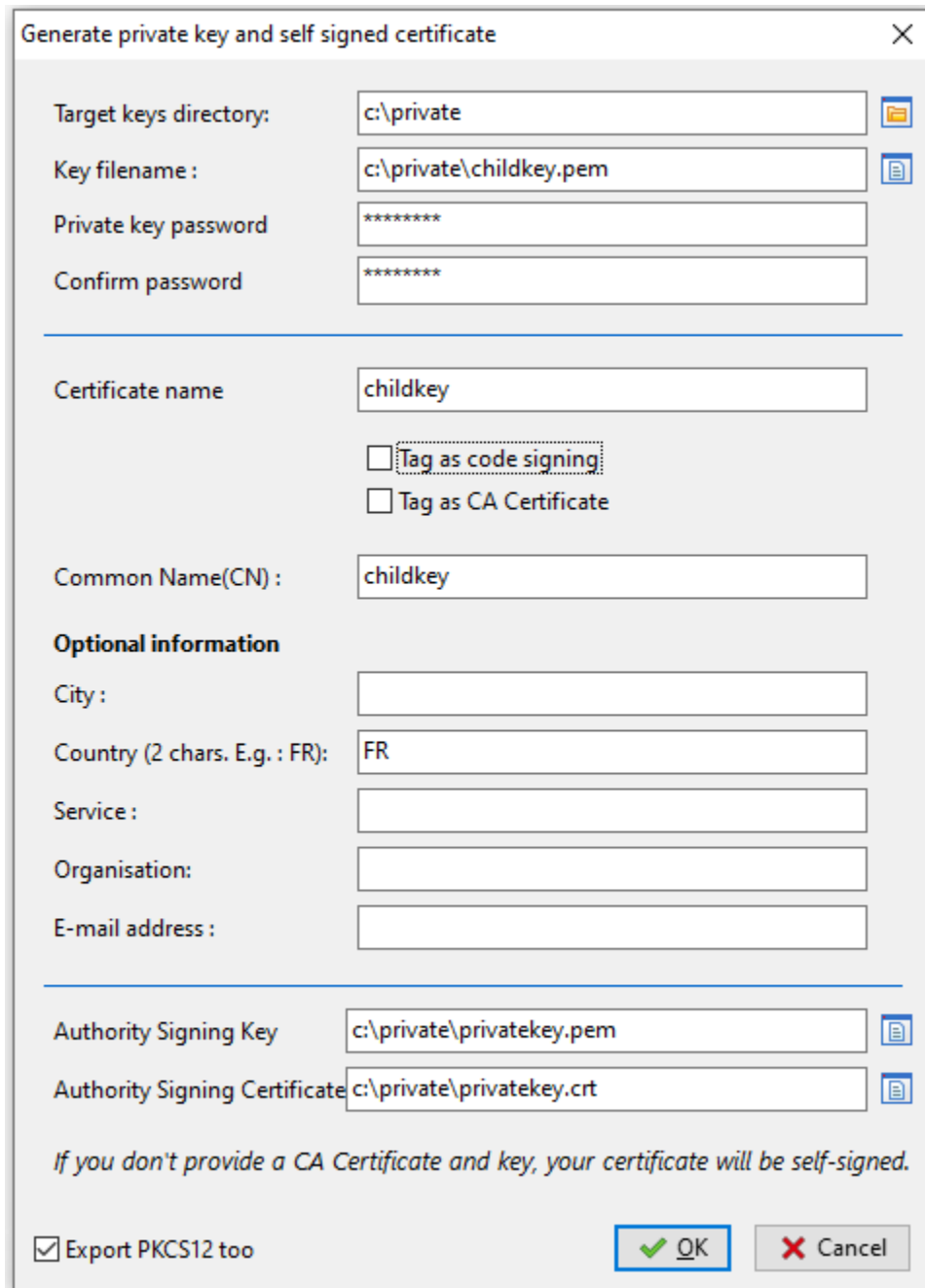
**Indication :** Les nouveaux certificats ne seront pas des *Autorités de Certification*, ce qui signifie qu'ils ne seront pas autorisés à signer d'autres certificats.

En règle générale, il n'y a qu'une seule paire pem / crt d'**Autorité de Certification** par *Organisation*.

---

**Attention :** Il n'est pas nécessaire de déployer des certificats enfants avec l'agent WAPT.

Les certificats enfants sont utilisés avec la console WAPT pour autoriser ou restreindre les actions dans la console.



Generate private key and self signed certificate

Target keys directory: c:\private

Key filename : c:\private\childkey.pem

Private key password: \*\*\*\*\*

Confirm password: \*\*\*\*\*

---

Certificate name: childkey

Tag as code signing

Tag as CA Certificate

Common Name(CN) : childkey

**Optional information**

City :

Country (2 chars. E.g. : FR): FR

Service :

Organisation:

E-mail address :

---

Authority Signing Key: c:\private\privatekey.pem

Authority Signing Certificate: c:\private\privatekey.crt

*If you don't provide a CA Certificate and key, your certificate will be self-signed.*

Export PKCS12 too

OK Cancel

FIG. 1 – Génération d'un certificat sans l'attribut *Pour Signature de code*

Generate private key and self signed certificate

Target keys directory: c:\private

Key filename : c:\private\childkey.pem

Private key password: \*\*\*\*\*

Confirm password: \*\*\*\*\*

---

Certificate name: childkey

Tag as code signing  
 Tag as CA Certificate

Common Name(CN) : childkey

**Optional information**

City:

Country (2 chars. E.g. : FR): FR

Service:

Organisation:

E-mail address:

---

Authority Signing Key: c:\private\privatekey.pem

Authority Signing Certificate: c:\private\privatekey.crt

*If you don't provide a CA Certificate and key, your certificate will be self-signed.*

Export PKCS12 too

OK Cancel

FIG. 2 – Génération d'un certificat avec l'attribut *Pour Signature de code*

## 21.7 Déploiement des certificats des administrateurs informatiques locaux sur les clients

**Indication :** Certaines organisations choisiront de laisser les administrateurs informatiques locaux effectuer des actions sur les appareils équipés de WAPT en leur délivrant des certificats personnels qui fonctionneront sur l'ensemble des appareils dont les administrateurs informatiques locaux sont responsables.

Les administrateurs informatiques du siège déploieront les certificats des administrateurs informatiques locaux sur les ordinateurs que les administrateurs locaux gèrent sur leurs sites respectifs.

Ainsi, les administrateurs informatiques locaux ne pourront pas gérer les ordinateurs situés au siège, mais uniquement sur leurs propres sites.

Il est possible de gérer simplement et de manière plus fine en utilisant *Access Control Lists* avec la version Enterprise de WAPT.

Vous devrez copier les certificats des administrateurs informatiques locaux autorisés sur les clients WAPT dans `C:\program files(x86)\wapt\ssl`.

**Indication :** N'oubliez pas de redémarrer le service WAPT sur les clients pour qu'ils utilisent leur nouveau certificat. Ouvrez une ligne de commande `cmd.exe` puis :

```
net stop waptservice && net start waptservice
```

Si vous voulez déployer les certificats en utilisant WAPT, utilisez un *paquet de certificat*

## 21.8 Configuration des listes de contrôle d'accès

**Indication :** L'utilisateur `admin` par défaut de WAPT est authentifié par mot de passe stocké dans `waptserver.ini` dans la valeur de la clé `wapt_password`. Les autres utilisateurs WAPT peuvent être des utilisateurs locaux (`htpasswd_path`) ou des comptes utilisateurs AD (`ldap_auth_server` / `ldap_auth_base_dn`).

Les ACL définissent les actions autorisées pour tous les types d'utilisateurs dans le contexte WAPT.

**Note :** Les ACLs par défaut au niveau utilisateur sont définies par `default_ldap_users_acls` dans `waptserver.ini`.

L'ACL par défaut pour un nouvel utilisateur est `vue`.

**Attention :** La sécurité est définie par le certificat déployé sur les clients, et non par les ACL.

Les ACL limitent simplement les actions que le serveur est autorisé à relayer de la console WAPT aux agents WAPT.

A la date du `|date|`, les agents WAPT ne vérifient pas les droits ACL.

Pour configurer les ACL dans WAPT, allez dans *Outils* → *Gérer les utilisateurs Wapt et des droits*.

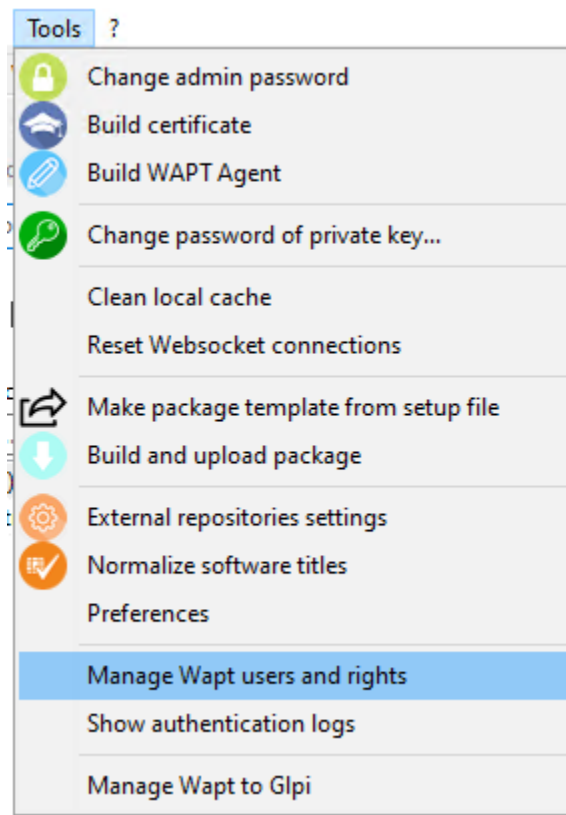


FIG. 3 – Gérer les utilisateurs et les droits WAPT dans l'onglet Outils

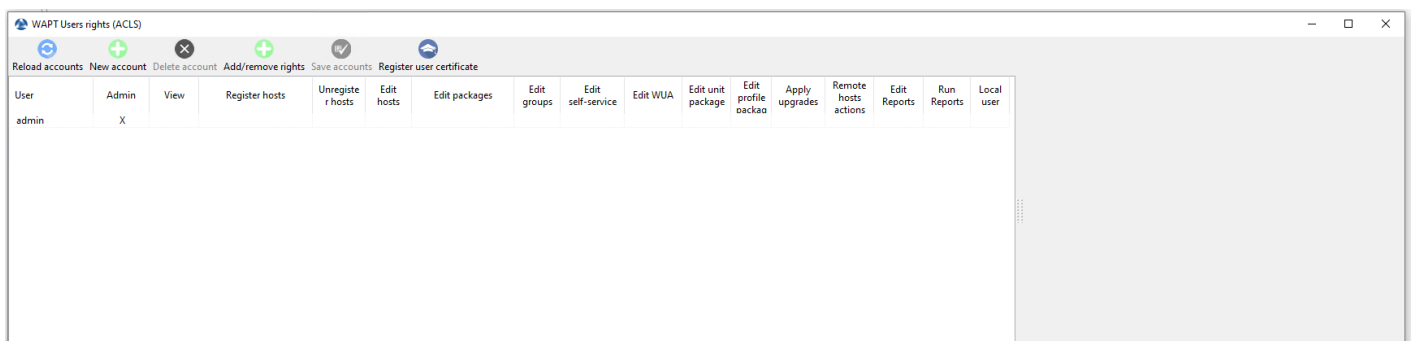


FIG. 4 – Gérer les utilisateurs et les droits WAPT au premier lancement

Au premier lancement après l'installation du serveur, seul le compte *SuperAdmin* est présent dans la liste des utilisateurs.

**Note :** Si le compte *SuperAdmin* n'existe pas ou ne possède pas le droit *admin*, le compte est recréé en redémarrant le service *waptserver*.

Le compte *SuperAdmin* est authentifié en utilisant la valeur de `wapt_password` dans le fichier de configuration `waptserver.ini`.

Deux types de comptes sont gérables par ACL, *local* et *Active Directory*.

## 21.8.1 Compte d'utilisateur local

Les utilisateurs locaux sont définis par un fichier `.htpasswd`.

### Configuration du serveur WAPT

Pour utiliser un compte utilisateur local, vous devez créer un fichier nommé `waptusers.htpasswd` dans le même *dossier* sur le serveur WAPT contenant le fichier `waptserver.ini`.

- Créez le `waptusers.htpasswd`.
- Sur Linux :

```
touch /opt/wapt/conf/waptusers.htpasswd
chown wapt /opt/wapt/conf/waptusers.htpasswd
```

- Sur Windows :

```
cd. > C:\wapt\conf\waptusers.htpasswd
```

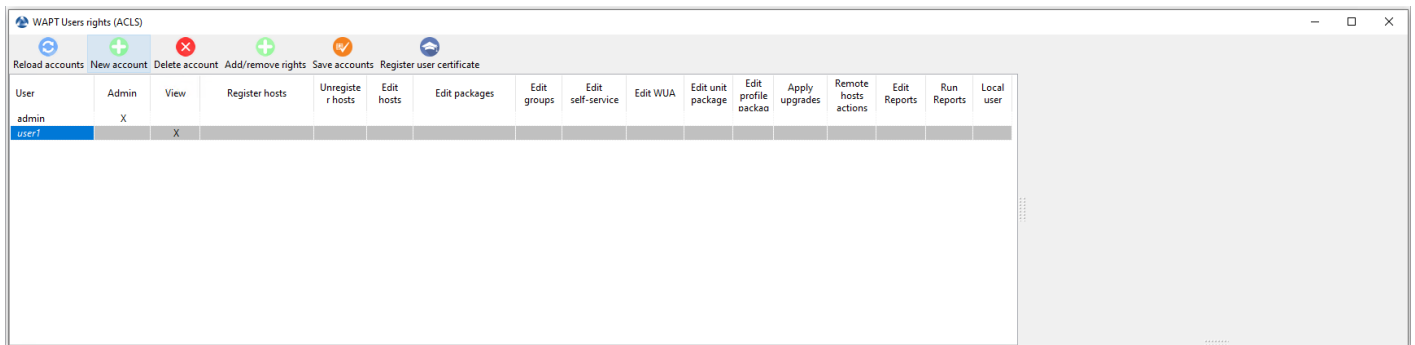
- Sur `waptserver.ini` ajoutez les paramètres `htpasswd_path`.

```
htpasswd_path = password file location
```

**Indication :** Redémarrer le service `waptserver`

### Création du compte utilisateur

- Dans la fenêtre *Droits des utilisateurs WAPT*, cliquez sur *Nouveau compte*.



Il est possible de renommer des comptes en appuyant sur F2 sur la colonne *User*.

- Sauvegardez en cliquant sur *Enregistrer les comptes*.
- Pour définir un mot de passe, voir le point **Changez le mot de passe**.
- Pour définir les droits, consultez la section *gérer les droits ACL*.

Si l'utilisateur local a un mot de passe dans `waptusers.htpasswd`, alors, le nom d'utilisateur apparaît en **gras** et *Mots de passe* est coché, sinon changez le mot de passe pour cet utilisateur.

### Changer le mot de passe de l'utilisateur

Pour changer le mot de passe du compte sélectionné :

- Faites un *clic droit* sur le compte → *Changer le mot de passe utilisateur sur le serveur Wapt*.

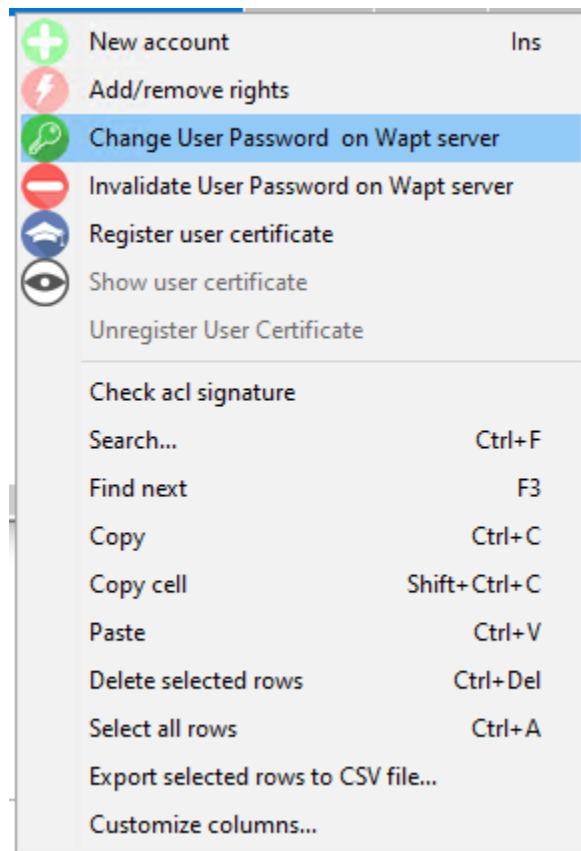


FIG. 5 – Changer le mot de passe de l'utilisateur

- Saisissez le nouveau mot de passe.

L'utilisateur local apparaît en *gras* et la case *Mots de passe* est cochée.



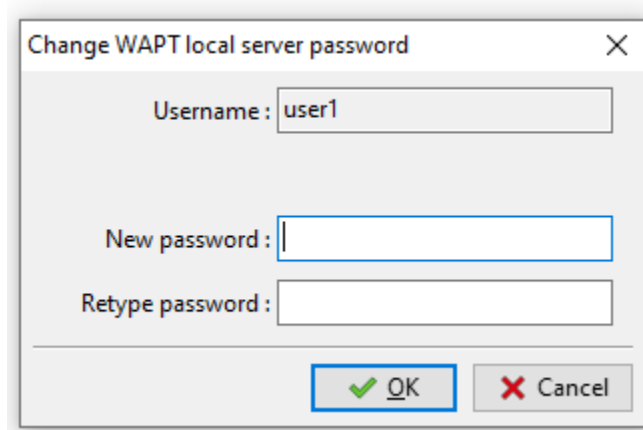


FIG. 6 – Saisir le nouveau mot de passe

### 21.8.2 Utilisateurs WAPT définis comme utilisateurs Active Directory

Pour gérer les utilisateurs WAPT avec votre Active Directory, vous devez activer l'*authentification Active Directory*. Après une première connexion réussie, le compte AD apparaîtra automatiquement dans la liste des utilisateurs WAPT.

### 21.8.3 Blocage des comptes d'utilisateurs locaux

Pour désenregistrer les utilisateurs locaux, faites *clic droit sur le compte* → *Invalider le mot de passe de l'utilisateur sur le serveur WAPT*.

Le compte sera bloqué et ne pourra plus gérer quoi que ce soit dans WAPT.

### 21.8.4 Liste des droits

De nombreux *droits et restrictions* peuvent être définis pour chaque utilisateur dans la console WAPT.

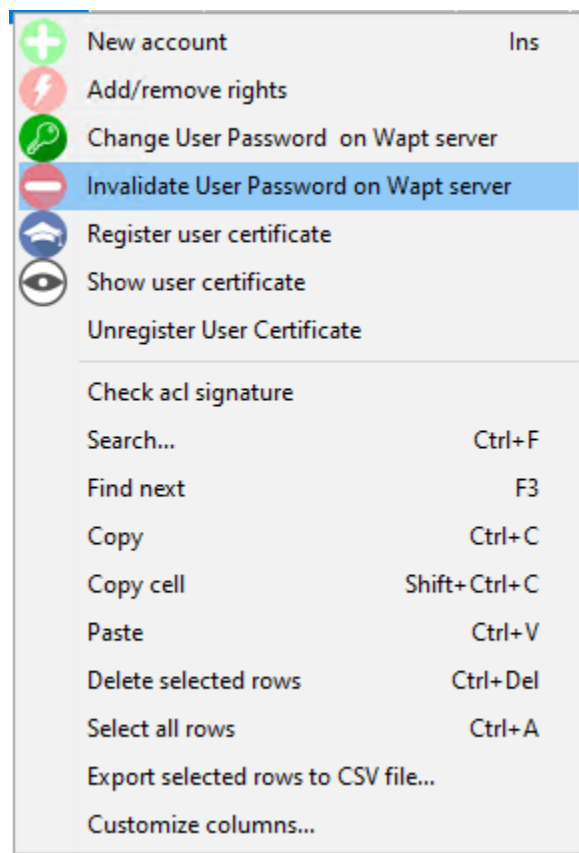


FIG. 7 – Invalidation d'un compte d'utilisateur local

TABLEAU 3 – Liste des droits des utilisateurs

Droit	Description
<i>Admin</i>	Comme SuperAdmin, tous les droits sont accordés sauf <i>Mot de passe</i> .
<i>Voir</i>	Permet de visualiser uniquement les informations sur la console WAPT.
<i>Inscrire machine</i>	Permet d'utiliser les informations d'identification Admin pour <i>enregistrer manuellement un hôte</i> avec le serveur WAPT.
<i>Désinscrire machine</i>	Permet de <i>supprimer une machine</i> depuis la console WAPT.
<i>Modif machine</i>	Permet de <i>modifier le paquet machine</i> sur la console WAPT.
<i>Modif paquets</i>	Permet de <i>modifier les paquets de base</i> qu'elle est autorisée à modifier.
<i>Modif groupes</i>	Permet de <i>modifier les paquets de groupe</i> sur la console WAPT.
<i>Modif self-service</i>	Permet de <i>modifier les règles de self-service</i> sur la console WAPT.
<i>Modif WUA</i>	Permet de <i>modifier les règles WUA / WSUS</i> sur la console WAPT.
<i>Modif paquets AD OU</i>	Permet de <i>modifier les paquets unit</i> sur la console WAPT.
<i>Modif paquets Profile</i>	Permet de <i>modifier les packages profile</i> sur la console WAPT.
<i>Lancer les installations</i>	Permet d'appliquer à distance des mises à jour sur son périmètre, si la machine est en statut <b>PENDING</b> .
<i>Actions distantes machine</i>	Permet d'utiliser les <i>outils Windows de Gestion de l'ordinateur</i> avec la console WAPT.
<i>Modifier requêtes</i>	Permet de <i>créer ou modifier des requêtes de rapport</i> .
<i>Lancer requête</i>	Permet de <i>exécuter des rapports SQL existants</i> .
<i>Mot de passe</i>	Définit un utilisateur local

## 21.8.5 Gestion des droits

Par défaut, le **SuperAdmin** est l'utilisateur du *Certificat CA*.

Pour les autres utilisateurs, il est possible d'associer un certificat qui a été généré à partir de la PKI WAPT ou d'une autre CA.

Ces certificats peuvent ou non être des enfants de l'autorité de certification WAPT.

**Attention :** Si les certificats ne sont pas émis par l'autorité de certification :

- Les paquets mis à jour sont disponibles uniquement sur les ordinateurs où les certificats sont déployés.
- Les ACL sont valides uniquement sur le périmètre des hôtes où le certificat de l'administrateur est déployé.

### Associer un certificat à un utilisateur

**Indication :** Par défaut, aucun certificat n'est défini pour aucun utilisateur (y compris *SuperAdmin*).

Le compte dans la console WAPT apparaît en *italic* si aucun certificat n'est associé à l'utilisateur.

Pour associer un certificat à un utilisateur, faites *clic droit sur l'utilisateur* → *Associer un certificat à l'utilisateur*.

Ensuite, choisissez le certificat à associer à l'utilisateur.

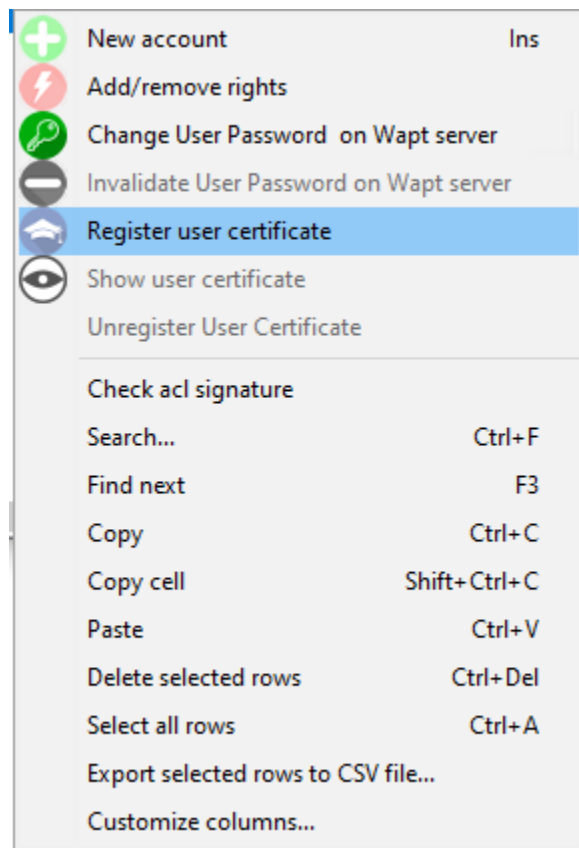


FIG. 8 – Enregistrement des certificats d'utilisateur

## Ajouter / supprimer des droits

Pour ajouter ou supprimer des droits, sélectionnez la cellule avec *clic gauche* et cochez-la en appuyant sur la barre d'espace.

**Indication :** Il est possible d'effectuer une sélection multiple en utilisant les raccourcis clavier *control+clic gauche* et en appuyant sur la barre d'espace.

## Restreindre le périmètre des droits accordés à l'utilisateur

Il est possible d'associer un périmètre à un droit donné à un utilisateur.

### Vue

TABLEAU 4 – Définition du périmètre autorisé

Périmètre	Description
<i>Tout refuser</i>	Aucun droit de regard n'est autorisé (non coché).
<i>Autoriser sur tout le périmètre</i>	La visualisation est autorisée pour tous les agents WAPT.
<i>Autoriser des périmètres spécifiques</i>	La visualisation est autorisée sur le périmètre sélectionné défini comme une liste de certificats.
<i>Autoriser où le certificat d'utilisateur est déployé</i>	La visualisation est autorisée uniquement sur le périmètre où le certificat de l'administrateur est déployé.

## Modifier les paquets de groupe

**Indication :** Tous les paquets de groupe fonctionnent sur le même principe, décrit ci-dessous.

TABLEAU 5 – Définition du périmètre autorisé

Périmètre	Description
<i>Interdire tous les paquets</i>	Aucune édition n'est autorisée pour aucun paquet (non coché).
<i>Autoriser tous les paquets</i>	Le droit de modification est autorisé pour tous les paquets.
<i>Autoriser des noms de paquets spécifiques</i>	Le droit de modification est autorisé pour les paquets sélectionnés disponibles dans la liste.

**Indication :** Si votre serveur WAPT est une machine virtuelle, prenez un instantané de la VM. De cette façon, vous pourrez revenir en arrière facilement dans le cas rare où la mise à jour échoue.

**Avertissement :** Après chaque mise à jour du serveur, mettez à jour votre *console* puis *régénérer* l'agent WAPT.

Avant de mettre à niveau le serveur WAPT, veuillez consulter le tableau de compatibilité de mise à niveau suivant :

TABLEAU 6 – Possibilités de mise à niveau WAPT disponibles

	Vers 1.5	WAPT	Vers 1.6	WAPT	Vers 1.7	WAPT	Vers 1.8	WAPT	Vers 2.0	WAPT	Vers 2.1	WAPT
Depuis WAPT 1.3	✓		✓		✗		✗		✗		✗	
Depuis WAPT 1.5	—		✓		✓		✗		✗		✗	
Depuis WAPT 1.6	—		—		✓		✗		✗		✗	
Depuis WAPT 1.7	—		—		—		✓		✗		✗	
Depuis WAPT 1.8.2	—		—		—		—		✓		✓	

---

### Mettre à jour de Community à Enterprise

---

Pour mettre à jour WAPT Community vers WAPT Enterprise suivez la méthode standard de *mise à jour de 1.8.2 vers 2.1*. Le serveur fera les changements appropriés.





## CHAPITRE 23

---

Mettre à jour de Discovery à Enterprise

---

---

**Note :** Non disponible à la date du 2024-01-09.

---



---

## Mise à jour mineure du serveur WAPT

---

### 24.1 Debian

**Indication :** Ce processus n'est valable que pour Debian (Pour ubuntu *ici*).

- Avant de mettre à jour, si ce n'est pas déjà fait, appliquez cette *configuration* à vos dépôts.
- S'il est correct, exécutez le bloc suivant.

```
export DEBIAN_FRONTEND=noninteractive
apt update
apt install tis-waptserver tis-waptsetup
unset DEBIAN_FRONTEND
```

- Puis exécutez le script **post-configuration**.

**Attention :** Pour que le post-configuration fonctionne correctement, vous devez d'abord avoir configuré correctement le *hostname* du serveur WAPT. Pour vérifier, utilisez la commande **echo \$(hostname)** qui doit vous renvoyer l'adresse DNS qui sera utilisé par les agents WAPT sur les ordinateurs clients.

**Avertissement :** Le script de post-configuration réécrit la configuration de nginx.

Si vous utilisez une *configuration spéciale*, sauvegardez votre fichier `wapt.conf` avec la commande

```
cp /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-available/wapt.conf.old
```

Il sera nécessaire d'écraser la configuration après la post-configuration avec la commande :

```
cp /etc/nginx/sites-available/wapt.conf.old /etc/nginx/sites-available/wapt.conf
```

**Indication :** Ce script de post-configuration doit être lancé en tant que **root**.

---

— Lancer le script.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
do you want to launch post configuration tool?  
  
      < yes >      < no >
```

— Choisissez un mot de passe (si ce n'est pas déjà défini) pour le compte *SuperAdmin* du serveur WAPT (la longueur minimale est de 10 caractères).

```
Please enter the wapt server password (min. 10 characters)
```

```
*****
```

```
      < OK >      < Cancel >
```

— Ne pas changer le mot de passe.

```
Please enter the server password again:
```

```
*****
```

```
      < OK >      < Cancel >
```

— Choisissez le mode d'authentification pour l'enregistrement initial des agents WAPT :

- Le choix #1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
- Le choix #2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
- Le choix #3 n'active pas le mécanisme d'authentification kerberos pour l'enregistrement initial des machines équipées de WAPT. Le serveur WAPT demandera un login et un mot de passe pour chaque machine s'enregistrant auprès de lui.

```
WaptAgent Authentication type?
```

- ```
-----  
(x) 1 Allow unauthenticated registration  
( ) 2 Enable kerberos authentication required for machines registration.  
      Registration will ask for password if kerberos not available  
( ) 3 Disable kerberos but registration require strong authentication  
-----
```

```
      < OK >      < Cancel >
```

— Sélectionnez *OK* pour démarrer le serveur WAPT.

```
Press OK to start waptserver
```

```
< OK >
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
Do you want to configure nginx?
```

```
< Yes >      < No >
```

— Renseignez le nom *FQDN* du serveur WAPT.

```
FQDN for the WAPT server (eg. wapt.example.com)
```

```
-----  
wapt.mydomain.lan  
-----
```

```
< OK >      < Cancel >
```

— Sélectionnez *OK* et un certificat auto-signé sera généré, cette étape peut prendre un certain temps.

```
Generating DH parameters, 2048 bit long safe prime, generator 2 This is going to take a long time  
.....+.....+....
```

Nginx est désormais configuré, sélectionnez *OK* pour redémarrer **Nginx** :

```
The Nginx config is done.  
We need to restart Nginx?
```

```
< OK >
```

Lancer l'étape de post-configuration.

```
Postconfiguration completed.  
Please connect to https://wapt.mydomain.lan/ to access the server.
```

```
< OK >
```

Puis exécutez le script de post-configuration :

| Options                    | Description                                                                                          |
|----------------------------|------------------------------------------------------------------------------------------------------|
| <code>--force-https</code> | Configurer <b>Nginx</b> de sorte que <i>le port 80 soit redirigé de façon permanente vers le 443</i> |

— Une fois terminé, votre serveur est prêt.

## 24.2 Ubuntu

- Avant de mettre à jour, si ce n'est pas déjà fait, appliquez cette *configuration* à vos dépôts.
- S'il est correct, exécutez le bloc suivant.

```
export DEBIAN_FRONTEND=noninteractive
apt update
apt install tis-waptserver tis-waptsetup
unset DEBIAN_FRONTEND
```

- Puis exécutez le script **post-configuration**.

**Attention :** Pour que le post-configuration fonctionne correctement, vous devez d'abord avoir configuré correctement le *hostname* du serveur WAPT. Pour vérifier, utilisez la commande **echo \$(hostname)** qui doit vous renvoyer l'adresse DNS qui sera utilisé par les agents WAPT sur les ordinateurs clients.

**Avertissement :** Le script de post-configuration réécrit la configuration de nginx.

Si vous utilisez une *configuration spéciale*, sauvegardez votre fichier `wapt.conf` avec la commande

```
cp /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-available/wapt.conf.old
```

Il sera nécessaire d'écraser la configuration après la post-configuration avec la commande :

```
cp /etc/nginx/sites-available/wapt.conf.old /etc/nginx/sites-available/wapt.conf
```

---

**Indication :** Ce script de post-configuration doit être lancé en tant que **root**.

---

- Lancer le script.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

- Cliquez sur *Terminer* pour fermer la fenêtre.

```
do you want to launch post configuration tool?
```

```
< yes >          < no >
```

- Choisissez un mot de passe (si ce n'est pas déjà défini) pour le compte *SuperAdmin* du serveur WAPT (la longueur minimale est de 10 caractères).

```
Please enter the wapt server password (min. 10 characters)
```

```
*****
```

```
< OK >          < Cancel >
```

- Ne pas changer le mot de passe.

Please enter the server password again:

\*\*\*\*\*

< OK >            < Cancel >

- Choisissez le mode d'authentification pour l'enregistrement initial des agents WAPT :
  - Le choix #1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
  - Le choix #2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
  - Le choix #3 n'active pas le mécanisme d'authentification kerberos pour l'enregistrement initial des machines équipées de WAPT. Le serveur WAPT demandera un login et un mot de passe pour chaque machine s'enregistrant auprès de lui.

WaptAgent Authentication type?

- 
- 1 Allow unauthenticated registration
  - 2 Enable kerberos authentication required **for** machines registration.  
Registration will ask **for** password **if** kerberos not available
  - 3 Disable kerberos but registration require strong authentication
- 

< OK >            < Cancel >

- Sélectionnez *OK* pour démarrer le serveur WAPT.

Press OK to start waptserver

< OK >

- Cliquez sur *Terminer* pour fermer la fenêtre.

Do you want to configure nginx?

< Yes >            < No >

- Renseignez le nom *FQDN* du serveur WAPT.

FQDN **for** the WAPT server (eg. wapt.example.com)

-----  
wapt.mydomain.lan  
-----

< OK >            < Cancel >

- Sélectionnez *OK* et un certificat auto-signé sera généré, cette étape peut prendre un certain temps.

Generating DH parameters, 2048 bit long safe prime, generator 2 This is going to take a long time  
.....+.....+.....

Nginx est désormais configuré, sélectionnez *OK* pour redémarrer **Nginx** :

```
The Nginx config is done.
We need to restart Nginx?
```

```
< OK >
```

Lancer l'étape de post-configuration.

```
Postconfiguration completed.
Please connect to https://wapt.mydomain.lan/ to access the server.
```

```
< OK >
```

Puis exécutez le script de post-configuration :

| Options                    | Description                                                                                          |
|----------------------------|------------------------------------------------------------------------------------------------------|
| <code>--force-https</code> | Configurer <b>Nginx</b> de sorte que <i>le port 80 soit redirigé de façon permanente vers le 443</i> |

— Une fois terminé, votre serveur est prêt.

## 24.3 Centos / RedHat

- Avant de mettre à jour, si ce n'est pas déjà fait, appliquez cette *configuration* à vos dépôts.
- S'il est correct, exécutez le bloc suivant.

```
yum update -y
```

— Installez les paquets.

```
yum install tis-waptserver tis-waptsetup -y
```

— Puis exécutez le script de **post-configuration**.

**Attention :** Pour que le post-configuration fonctionne correctement, vous devez d'abord avoir configuré correctement le *hostname* du serveur WAPT. Pour vérifier, utilisez la commande **echo \$(hostname)** qui doit vous renvoyer l'adresse DNS qui sera utilisé par les agents WAPT sur les ordinateurs clients.

**Avertissement :** Le script de post-configuration réécrit la configuration de nginx.

Si vous utilisez une *configuration spéciale*, sauvegardez votre fichier `wapt.conf` avec la commande

```
cp /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-available/wapt.conf.old
```

Il sera nécessaire d'écraser la configuration après la post-configuration avec la commande :

```
cp /etc/nginx/sites-available/wapt.conf.old /etc/nginx/sites-available/wapt.conf
```



**Indication :** Ce script de post-configuration doit être lancé en tant que **root**.

— Lancer le script.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
do you want to launch post configuration tool?
```

```
< yes >          < no >
```

— Choisissez un mot de passe (si ce n'est pas déjà défini) pour le compte *SuperAdmin* du serveur WAPT (la longueur minimale est de 10 caractères).

```
Please enter the wapt server password (min. 10 characters)
```

```
*****
```

```
< OK >          < Cancel >
```

— Ne pas changer le mot de passe.

```
Please enter the server password again:
```

```
*****
```

```
< OK >          < Cancel >
```

— Choisissez le mode d'authentification pour l'enregistrement initial des agents WAPT :

- Le choix #1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
- Le choix #2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
- Le choix #3 n'active pas le mécanisme d'authentification kerberos pour l'enregistrement initial des machines équipées de WAPT. Le serveur WAPT demandera un login et un mot de passe pour chaque machine s'enregistrant auprès de lui.

```
WaptAgent Authentication type?
```

- ```
-----
(x) 1 Allow unauthenticated registration
( ) 2 Enable kerberos authentication required for machines registration.
    Registration will ask for password if kerberos not available
( ) 3 Disable kerberos but registration require strong authentication
-----
```

```
< OK >          < Cancel >
```

— Sélectionnez *OK* pour démarrer le serveur WAPT.

```
Press OK to start waptserver
```

```
< OK >
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
Do you want to configure nginx?  
  
    < Yes >      < No >
```

— Renseignez le nom *FQDN* du serveur WAPT.

```
FQDN for the WAPT server (eg. wapt.example.com)  
-----  
wapt.mydomain.lan  
-----  
  
    < OK >      < Cancel >
```

— Sélectionnez *OK* et un certificat auto-signé sera généré, cette étape peut prendre un certain temps.

```
Generating DH parameters, 2048 bit long safe prime, generator 2 This is going to take a long time  
.....+.....+...
```

Nginx est désormais configuré, sélectionnez *OK* pour redémarrer **Nginx** :

```
The Nginx config is done.  
We need to restart Nginx?  
  
    < OK >
```

Lancer l'étape de post-configuration.

```
Postconfiguration completed.  
Please connect to https://wapt.mydomain.lan/ to access the server.  
  
    < OK >
```

Puis exécutez le script de post-configuration :

Options	Description
<code>--force-https</code>	Configurer <b>Nginx</b> de sorte que <i>le port 80 soit redirigé de façon permanente vers le 443</i>

— Une fois terminé, votre serveur est prêt.

## 24.4 Windows

### 24.4.1 Discovery

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition **WAPT Discovery**. Pour **WAPT Enterprise** Edition, veuillez vous référer au bloc suivant.

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>.

### 24.4.2 Enterprise

**Indication :** Pour accéder aux ressources de **WAPT Enterprise**, vous devez utiliser le nom d'utilisateur et le mot de passe fournis par notre service commercial.

— Téléchargez et exécutez `waptserversetup.exe`.

**Attention :** L'installation du serveur WAPT doit être effectuée à l'aide d'un compte **Administrateur local** sur l'hôte

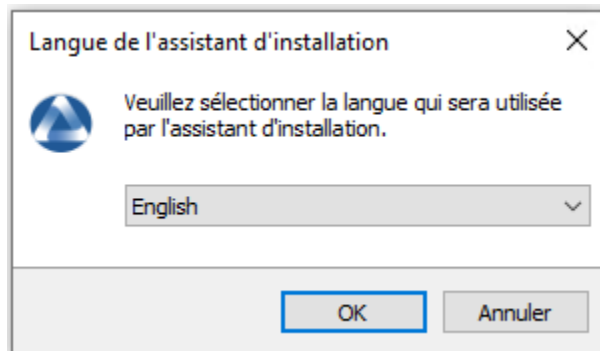
**Avertissement :** Le script de post-configuration réécrit la configuration de nginx. Si vous utilisez une configuration spéciale, sauvegardez votre fichier `nginx.conf` avec la commande :

```
copy C:\wapt\waptserver\nnginx\conf\nnginx.conf C:\wapt\waptserver\nnginx\conf\nnginx.conf.old
```

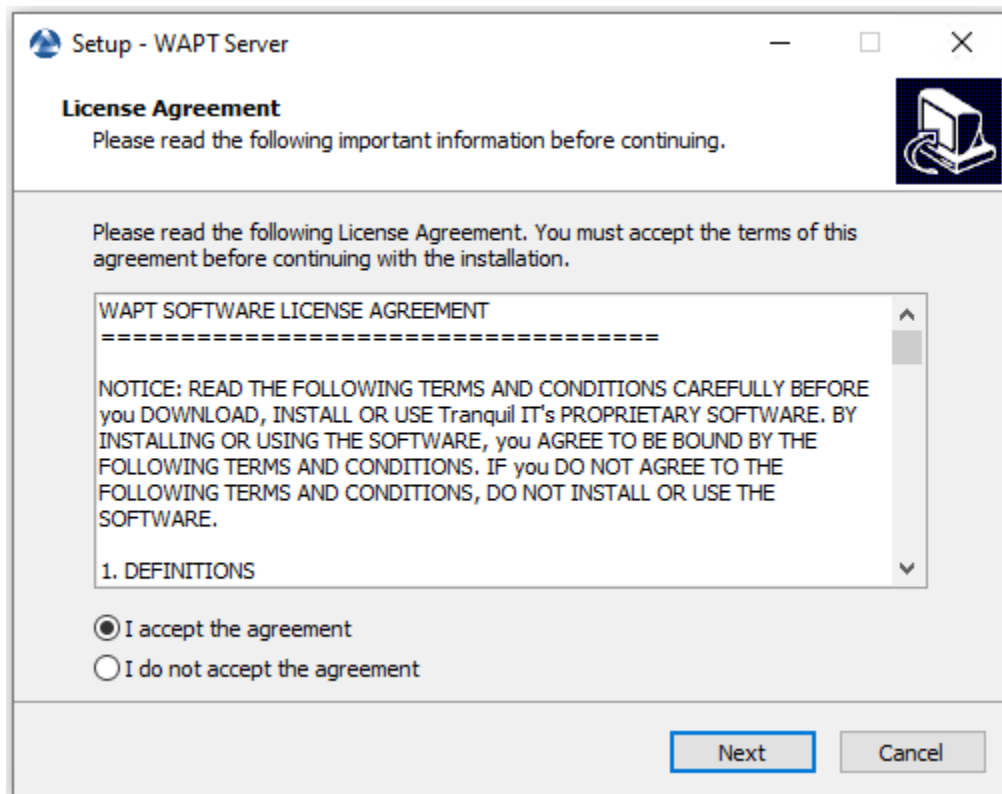
Il sera nécessaire d'écraser la configuration après la post-configuration avec la commande :

```
copy C:\wapt\waptserver\nnginx\conf\nnginx.conf.old C:\wapt\waptserver\nnginx\conf\nnginx.conf
```

— Choisissez la langue d'installation.

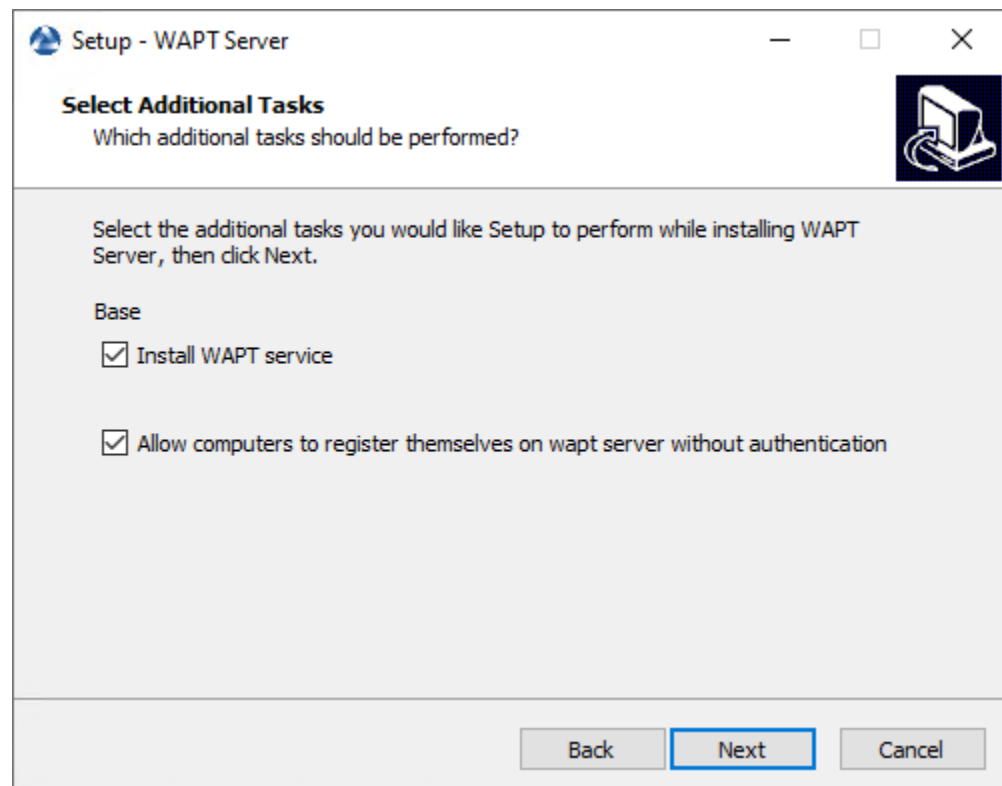
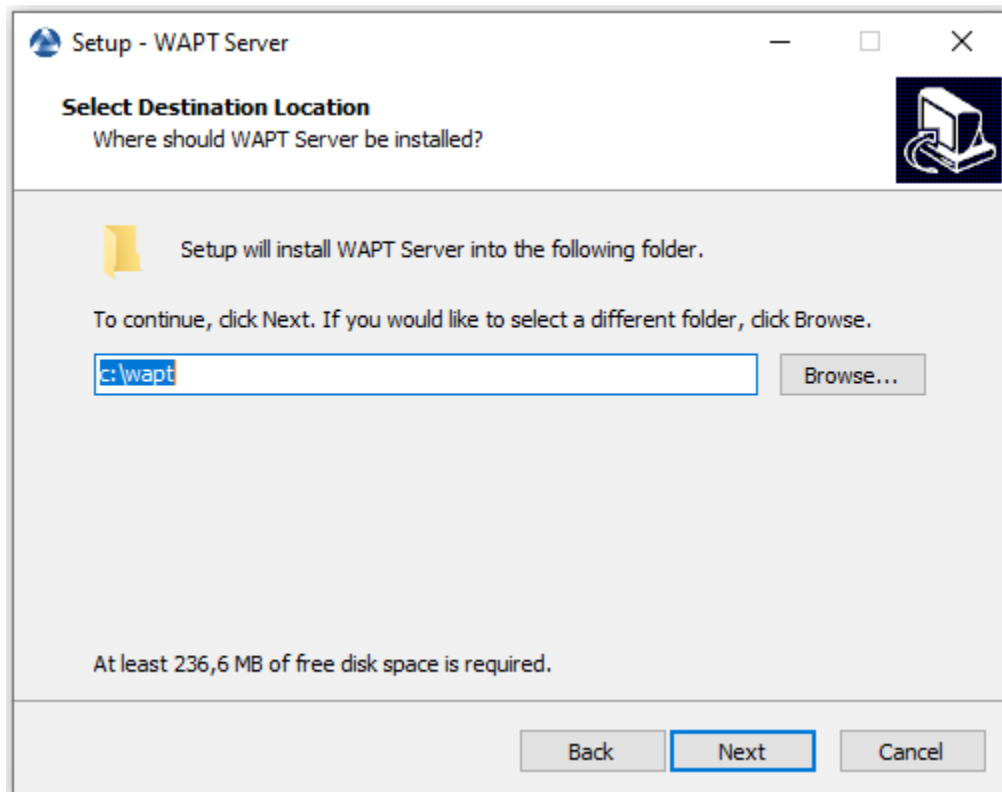


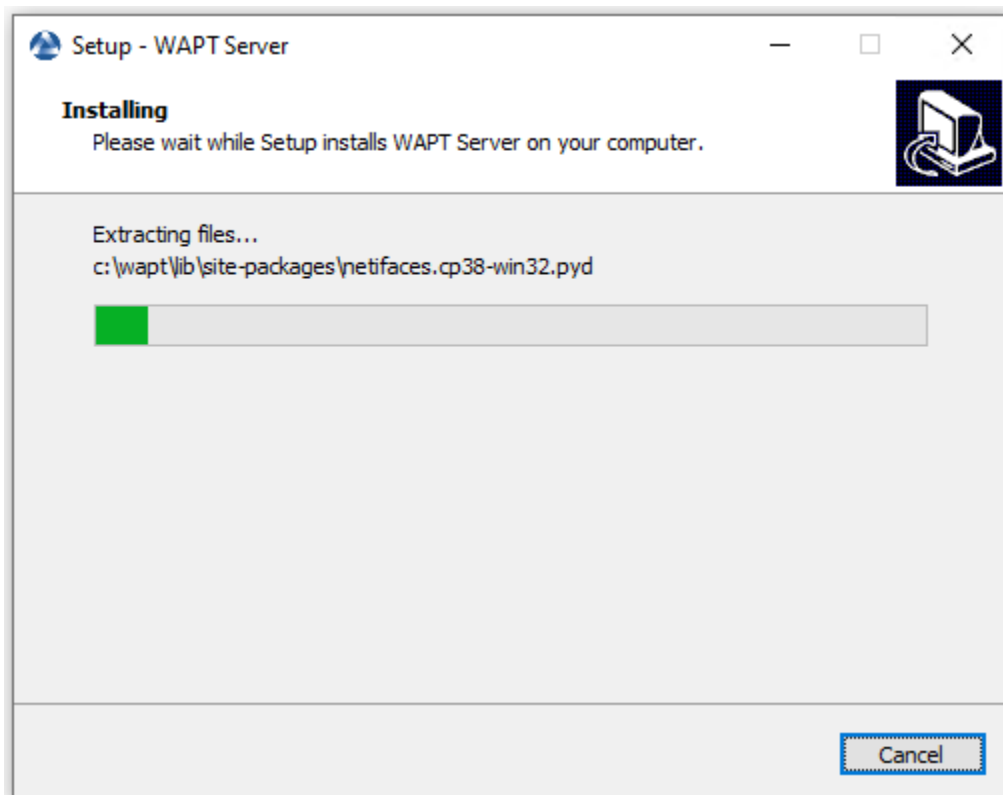
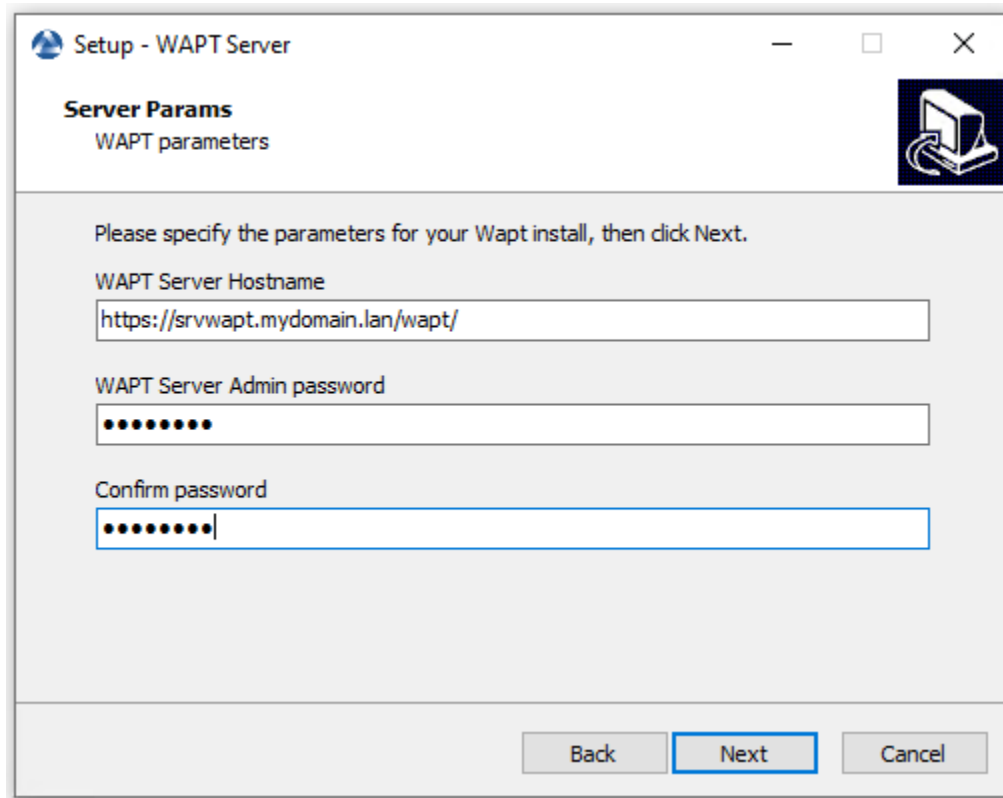
- Acceptez la licence publique GNU et cliquez sur *Suivant* pour passer à l'étape suivante.

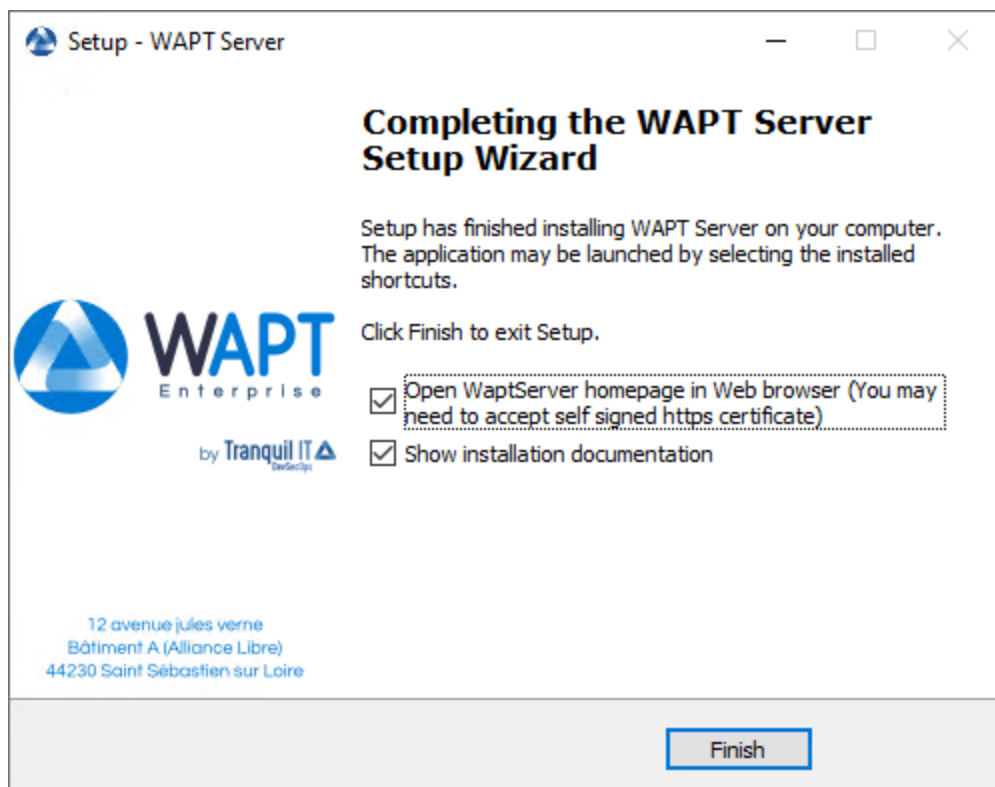


- Choisissez le répertoire d'installation (laissez la valeur par défaut) et cliquez sur *Suivant* pour passer à l'étape suivante.
- Choisissez une tâche supplémentaire (laissez la valeur par défaut).
- Ne pas modifier le mot de passe du serveur WAPT (si cela n'est pas nécessaire).
- Cliquez sur le bouton *Installer* pour lancer l'installation, attendez que l'installation soit terminée.
- Cliquez sur *Terminer* pour fermer la fenêtre.

Le serveur WAPT sur votre Windows est prêt.











---

## Mettre à jour de WAPT de 2.0 à 2.1

---

---

**Note :** Avant de procéder à la mise à niveau, lisez les *exigences d'installation*.

---

**Attention :** Les ports 80 et 443 sont utilisés par le serveur WAPT et doivent être disponibles.

### 25.1 Debian

---

**Note :** Si vous êtes sous Debian 9 Stretch, vous devez d'abord mettre à niveau vers Debian 10 Buster avant de mettre à niveau vers WAPT 2.X. **Le serveur WAPT 2.X n'est pas disponible pour Debian 9.**

---

— Tout d'abord, mettez à jour la distribution.

```
apt update && apt upgrade -y
```

— Ajouter ou mettre à jour le dépôt de paquets Debian, importer la clé GPG du dépôt et installer les paquets du serveur WAPT.

### 25.1.1 De Community à Discovery

---

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>

---

### 25.1.2 Enterprise

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

---

— Installez `apt-transport-https` pour l'utilisation de `https`.

```
apt install apt-transport-https lsb-release gnupg
```

— Récupération de la clé `.gpg` et ajout dans le dépôt Tranquil IT.

```
wget -O - https://wapt.tranquil.it/debian/tiswapt-pub.gpg | apt-key add -  
echo "deb https://srvwapt-pro.tranquil.it/entreprise/debian/wapt-2.1/ $(lsb_release -c -s) main" > /  
↪etc/apt/sources.list.d/wapt.list
```

**Note :**

— S'il n'existe pas, créez `wapt.conf` dans `/etc/apt/auth.conf.d` pour stocker vos informations de connexion.

---

**Indication :** Remplacez `user` et `password` pour accéder au dépôt WAPT Enterprise, ce dernier est fourni pour notre département de vente.

---

```
cat > /etc/apt/auth.conf.d/wapt.conf <<EOF  
machine srvwapt-pro.tranquil.it  
login user  
password password  
EOF
```

— Appliquer les bons droits :

```
chmod 600 /etc/apt/auth.conf.d/wapt.conf
```

— Mettre à jour le dépôt et installer les paquets.

```
apt update  
apt install tis-waptserver tis-waptsetup
```

### 25.1.3 Lancer l'étape de post-configuration

**Avertissement :** Sur WAPT 2.1 l'*activation de la licence* est modifiée. Il est nécessaire d'exécuter le script de post-configuration.

— Lancer l'étape de post-configuration.

**Attention :** Pour que le post-configuration fonctionne correctement, vous devez d'abord avoir configuré correctement le *hostname* du serveur WAPT. Pour vérifier, utilisez la commande **echo \$(hostname)** qui doit vous renvoyer l'adresse DNS qui sera utilisé par les agents WAPT sur les ordinateurs clients.

**Avertissement :** Le script de post-configuration réécrit la configuration de nginx.

Si vous utilisez une *configuration spéciale*, sauvegardez votre fichier `wapt.conf` avec la commande

```
cp /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-available/wapt.conf.old
```

Il sera nécessaire d'écraser la configuration après la post-configuration avec la commande :

```
cp /etc/nginx/sites-available/wapt.conf.old /etc/nginx/sites-available/wapt.conf
```

**Indication :** Ce script de post-configuration doit être lancé en tant que **root**.

— Lancer le script.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
do you want to launch post configuration tool?
```

```
< yes >          < no >
```

— Choisissez un mot de passe (si ce n'est pas déjà défini) pour le compte *SuperAdmin* du serveur WAPT (la longueur minimale est de 10 caractères).

```
Please enter the wapt server password (min. 10 characters)
```

```
*****
```

```
< OK >          < Cancel >
```

— Ne pas changer le mot de passe.

```
Please enter the server password again:
```

```
*****
```

(suite sur la page suivante)

(suite de la page précédente)

< OK >                      < Cancel >

- Choisissez le mode d'authentification pour l'enregistrement initial des agents WAPT :
  - Le choix #1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
  - Le choix #2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
  - Le choix #3 n'active pas le mécanisme d'authentification kerberos pour l'enregistrement initial des machines équipées de WAPT. Le serveur WAPT demandera un login et un mot de passe pour chaque machine s'enregistrant auprès de lui.

```
WaptAgent Authentication type?
-----
(x) 1 Allow unauthenticated registration
( ) 2 Enable kerberos authentication required for machines registration.
    Registration will ask for password if kerberos not available
( ) 3 Disable kerberos but registration require strong authentication
-----
                                < OK >                      < Cancel >
```

- Sélectionnez *OK* pour démarrer le serveur WAPT.

```
Press OK to start waptserver
                                < OK >
```

- Cliquez sur *Terminer* pour fermer la fenêtre.

```
Do you want to configure nginx?
                                < Yes >                      < No >
```

- Renseignez le nom *FQDN* du serveur WAPT.

```
FQDN for the WAPT server (eg. wapt.example.com)
-----
wapt.mydomain.lan
-----
                                < OK >                      < Cancel >
```

- Sélectionnez *OK* et un certificat auto-signé sera généré, cette étape peut prendre un certain temps.

```
Generating DH parameters, 2048 bit long safe prime, generator 2 This is going to take a long time
.....+.....+.....
```

Ngix est désormais configuré, sélectionnez *OK* pour redémarrer **Ngix** :

```
The Ngix config is done.
We need to restart Ngix?
```

(suite sur la page suivante)

(suite de la page précédente)

&lt; OK &gt;

Lancer l'étape de post-configuration.

```
Postconfiguration completed.
Please connect to https://wapt.mydomain.lan/ to access the server.
```

&lt; OK &gt;

Puis exécutez le script de post-configuration :

Options	Description
<code>--force-https</code>	Configurer <b>Nginx</b> de sorte que <i>le port 80 soit redirigé de façon permanente vers le 443</i>

— Redémarrez le serveur WAPT.

```
systemctl restart waptserver nginx
```

— Assurez-vous que le propriétaire du fichier est correct.

```
chown wapt:www-data -R /var/www/wapt*
```

## 25.2 Ubuntu

— Tout d'abord, mettez à jour la distribution.

```
apt update && apt upgrade -y
```

— Ajouter ou mettre à jour le dépôt de paquets Debian, importer la clé GPG du dépôt et installer les paquets du serveur WAPT.

### 25.2.1 De Community à Discovery

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>

## 25.2.2 Enterprise

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

---

— Installez `apt-transport-https` pour l'utilisation de `https`.

```
apt install apt-transport-https lsb-release gnupg
```

— Récupération de la clé `.gpg` et ajout dans le dépôt Tranquil IT.

```
wget -O - https://wapt.tranquil.it/debian/tiswapt-pub.gpg | apt-key add -  
echo "deb https://srvwapt-pro.tranquil.it/entreprise/ubuntu/wapt-2.1/ $(lsb_release -c -s) main" > /  
→etc/apt/sources.list.d/wapt.list
```

**Note :**

— S'il n'existe pas, créez `wapt.conf` dans `/etc/apt/auth.conf.d` pour stocker vos informations de connexion.

---

**Indication :** Remplacez **user** et **password** pour accéder au dépôt WAPT Enterprise, ce dernier est fourni pour notre département de vente.

---

```
cat > /etc/apt/auth.conf.d/wapt.conf <<EOF  
machine srvwapt-pro.tranquil.it  
login user  
password password  
EOF
```

— Appliquer les bons droits :

```
chmod 600 /etc/apt/auth.conf.d/wapt.conf
```

— Mettre à jour le dépôt et installer les paquets.

```
apt update  
apt install tis-waptserver tis-waptsetup
```

## 25.2.3 Lancer l'étape de post-configuration

**Avertissement :** Sur WAPT 2.1 *l'activation de la licence* est modifiée. Il est nécessaire d'exécuter le script de post-configuration.

— Lancer l'étape de post-configuration.

**Attention :** Pour que le post-configuration fonctionne correctement, vous devez d'abord avoir configuré correctement le *hostname* du serveur WAPT. Pour vérifier, utilisez la commande **echo \$(hostname)** qui doit vous renvoyer l'adresse DNS qui sera utilisé par les agents WAPT sur les ordinateurs clients.

**Avertissement :** Le script de post-configuration réécrit la configuration de nginx.

Si vous utilisez une *configuration spéciale*, sauvegardez votre fichier `wapt.conf` avec la commande

```
cp /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-available/wapt.conf.old
```

Il sera nécessaire d'écraser la configuration après la post-configuration avec la commande :

```
cp /etc/nginx/sites-available/wapt.conf.old /etc/nginx/sites-available/wapt.conf
```

**Indication :** Ce script de post-configuration doit être lancé en tant que **root**.

— Lancer le script.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
do you want to launch post configuration tool?
< yes >          < no >
```

— Choisissez un mot de passe (si ce n'est pas déjà défini) pour le compte *SuperAdmin* du serveur WAPT (la longueur minimale est de 10 caractères).

```
Please enter the wapt server password (min. 10 characters)
```

```
*****
```

```
< OK >          < Cancel >
```

— Ne pas changer le mot de passe.

```
Please enter the server password again:
```

```
*****
```

```
< OK >          < Cancel >
```

— Choisissez le mode d'authentification pour l'enregistrement initial des agents WAPT :

- Le choix #1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
- Le choix #2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
- Le choix #3 n'active pas le mécanisme d'authentification kerberos pour l'enregistrement initial des machines équipées de WAPT. Le serveur WAPT demandera un login et un mot de passe pour chaque machine s'enregistrant auprès de lui.

```
WaptAgent Authentication type?
-----
(x) 1 Allow unauthenticated registration
( ) 2 Enable kerberos authentication required for machines registration.
    Registration will ask for password if kerberos not available
( ) 3 Disable kerberos but registration require strong authentication
-----
                                < OK >      < Cancel >
```

— Sélectionnez *OK* pour démarrer le serveur WAPT.

```
Press OK to start waptserver
                                < OK >
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
Do you want to configure nginx?
                                < Yes >      < No >
```

— Renseignez le nom *FQDN* du serveur WAPT.

```
FQDN for the WAPT server (eg. wapt.example.com)
-----
wapt.mydomain.lan
-----
                                < OK >      < Cancel >
```

— Sélectionnez *OK* et un certificat auto-signé sera généré, cette étape peut prendre un certain temps.

```
Generating DH parameters, 2048 bit long safe prime, generator 2 This is going to take a long time
.....+.....+....
```

Nginx est désormais configuré, sélectionnez *OK* pour redémarrer **Nginx** :

```
The Nginx config is done.
We need to restart Nginx?
                                < OK >
```

Lancer l'étape de post-configuration.

```
Postconfiguration completed.
Please connect to https://wapt.mydomain.lan/ to access the server.
                                < OK >
```

Puis exécutez le script de post-configuration :



Options	Description
<code>--force-https</code>	Configurer <b>Nginx</b> de sorte que <i>le port 80 soit redirigé de façon permanente vers le 443</i>

— Redémarrez le serveur WAPT.

```
systemctl restart waptserver nginx
```

— Assurez-vous que le propriétaire du fichier est correct.

```
chown wapt:www-data -R /var/www/wapt*
```

## 25.3 CentOS / RedHat

— Tout d’abord, mettez à jour la distribution.

```
yum update
```

— Ajouter ou mettre à jour le dépôt de paquets Debian, importer la clé GPG du dépôt et installer les paquets du serveur WAPT.

### 25.3.1 CentOS7

#### De Community à Discovery

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l’édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

---

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>.

---

#### Enterprise

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l’édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

Pour accéder au site de téléchargement WAPT Enterprise, vous devez utiliser le nom d’utilisateur et le mot de passe fournis par notre service commercial.

---

— Ajout du dépôt Tranquil IT.

```
cat > /etc/yum.repos.d/wapt.repo <<EOF
[wapt]
name=WAPT Server Repo
baseurl=https://srvwapt-pro.tranquil.it/entreprise/centos7/wapt-2.1/
enabled=1
gpgcheck=1
EOF
```

### Installez tous les paquets nécessaires

— Récupération de la clé .gpg.

```
wget -q -O /tmp/tranquil_it.gpg "https://wapt.tranquil.it/centos7/RPM-GPG-KEY-TISWAPT-7"; rpm --
↳import /tmp/tranquil_it.gpg
```

— Installez tous les paquets nécessaires

```
yum install epel-release -y
yum install postgresql96-server postgresql96-contrib tis-waptserver tis-waptsetup cabextract -y
```

## 25.3.2 CentOS8

### De Community à Discovery

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

---

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>.

---

### Enterprise

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

Pour accéder au site de téléchargement WAPT Enterprise, vous devez utiliser le nom d'utilisateur et le mot de passe fournis par notre service commercial.

---

— Ajout du dépôt Tranquil iT

```
cat > /etc/yum.repos.d/wapt.repo <<EOF
[wapt]
name=WAPT Server Repo
baseurl=https://srvwapt-pro.tranquil.it/entreprise/centos8/wapt-2.1/
enabled=1
gpgcheck=1
EOF
```

### Installez tous les paquets nécessaires

— Récupération de la clé .gpg.

```
wget -q -O /tmp/tranquil_it.gpg "https://wapt.tranquil.it/centos7/RPM-GPG-KEY-TISWAPT-7"; rpm --
↳import /tmp/tranquil_it.gpg
```

— Installez tous les paquets nécessaires

```
yum install epel-release -y
yum install postgresql96-server postgresql96-contrib tis-waptserver tis-waptsetup cabextract -y
```

### 25.3.3 Lancer l'étape de post-configuration

**Avertissement :** Sur WAPT 2.1 l'activation de la licence est modifiée. Il est nécessaire d'exécuter le script de post-configuration.

**Attention :** Pour que le post-configuration fonctionne correctement, vous devez d'abord avoir configuré correctement le *hostname* du serveur WAPT. Pour vérifier, utilisez la commande **echo \$(hostname)** qui doit vous renvoyer l'adresse DNS qui sera utilisé par les agents WAPT sur les ordinateurs clients.

**Avertissement :** Le script de post-configuration réécrit la configuration de nginx.

Si vous utilisez une *configuration spéciale*, sauvegardez votre fichier `wapt.conf` avec la commande

```
cp /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-available/wapt.conf.old
```

Il sera nécessaire d'écraser la configuration après la post-configuration avec la commande :

```
cp /etc/nginx/sites-available/wapt.conf.old /etc/nginx/sites-available/wapt.conf
```

**Indication :** Ce script de post-configuration doit être lancé en tant que **root**.

— Lancer le script.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
do you want to launch post configuration tool?
```

```
< yes >          < no >
```

— Choisissez un mot de passe (si ce n'est pas déjà défini) pour le compte *SuperAdmin* du serveur WAPT (la longueur minimale est de 10 caractères).

```
Please enter the wapt server password (min. 10 characters)
```

```
*****
```

```
< OK >          < Cancel >
```

— Ne pas changer le mot de passe.

```
Please enter the server password again:
```

```
*****
```

```
< OK >          < Cancel >
```

— Choisissez le mode d'authentification pour l'enregistrement initial des agents WAPT :

- Le choix #1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
- Le choix #2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
- Le choix #3 n'active pas le mécanisme d'authentification kerberos pour l'enregistrement initial des machines équipées de WAPT. Le serveur WAPT demandera un login et un mot de passe pour chaque machine s'enregistrant auprès de lui.

```
WaptAgent Authentication type?
```

- ```
-----  
(x) 1 Allow unauthenticated registration  
( ) 2 Enable kerberos authentication required for machines registration.  
    Registration will ask for password if kerberos not available  
( ) 3 Disable kerberos but registration require strong authentication  
-----
```

```
< OK >          < Cancel >
```

— Sélectionnez *OK* pour démarrer le serveur WAPT.

```
Press OK to start waptserver
```

```
< OK >
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
Do you want to configure nginx?
```

```
< Yes >      < No >
```

— Renseignez le nom *FQDN* du serveur WAPT.

```
FQDN for the WAPT server (eg. wapt.example.com)
```

```
-----  
wapt.mydomain.lan  
-----
```

```
< OK >      < Cancel >
```

— Sélectionnez *OK* et un certificat auto-signé sera généré, cette étape peut prendre un certain temps.

```
Generating DH parameters, 2048 bit long safe prime, generator 2 This is going to take a long time  
.....+.....+....
```

Nginx est désormais configuré, sélectionnez *OK* pour redémarrer **Nginx** :

```
The Nginx config is done.  
We need to restart Nginx?
```

```
< OK >
```

Lancer l'étape de post-configuration.

```
Postconfiguration completed.  
Please connect to https://wapt.mydomain.lan/ to access the server.
```

```
< OK >
```

Puis exécutez le script de post-configuration :

| Options       | Description                                                                                          |
|---------------|------------------------------------------------------------------------------------------------------|
| --force-https | Configurer <b>Nginx</b> de sorte que <i>le port 80 soit redirigé de façon permanente vers le 443</i> |

— Redémarrez le serveur WAPT.

```
systemctl restart waptserver nginx
```

— Assurez-vous que le propriétaire du fichier est correct.

```
chown wapt:www-data -R /var/www/wapt*
```

## 25.4 Windows

### 25.4.1 De Community à Discovery

---

**Note :** Non disponible à la date du 2024-01-09.

---

### 25.4.2 Enterprise

---

**Indication :** Pour accéder aux ressources de WAPT Enterprise, vous devez utiliser le nom d'utilisateur et le mot de passe fournis par notre service commercial.

---

- Téléchargez et exécutez `waptserversetup.exe`.
- Choisissez la langue d'installation.

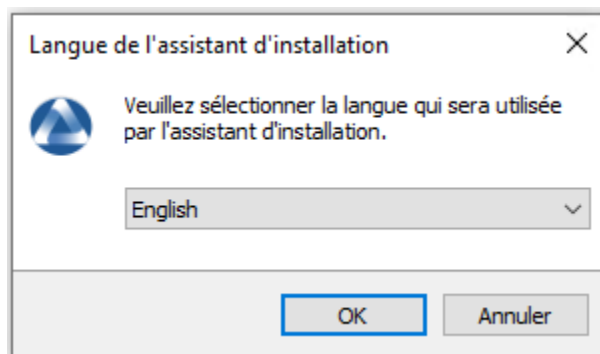


FIG. 1 – Choix de la langue pour le WAPT

- Acceptez la licence publique GNU et cliquez sur *Suivant* pour passer à l'étape suivante.
- Choisissez le répertoire d'installation (laissez la valeur par défaut) et cliquez sur *Suivant* pour passer à l'étape suivante.
- Choisissez le répertoire d'installation (laissez la valeur par défaut) et cliquez sur *Suivant* pour passer à l'étape suivante.
- Sélectionnez une tâche supplémentaire si nécessaire.
- Modifiez le mot de passe du serveur WAPT si nécessaire, puis appuyez sur *Suivant*.
- Cliquez sur le bouton *Installer* pour lancer l'installation, attendez que l'installation soit terminée.
- Cliquez sur *Terminer* pour fermer la fenêtre.

**Attention :** NE PAS utiliser la console WAPT sur le serveur WAPT. N'installez PAS et n'exécutez pas vos outils de développement de paquets WAPT sur le serveur WAPT.

Le serveur WAPT sur votre serveur ou station de travail Windows est prêt.

Votre serveur est maintenant prêt. Vous pouvez maintenant consulter la documentation sur *Installation de la console de gestion WAPT*.

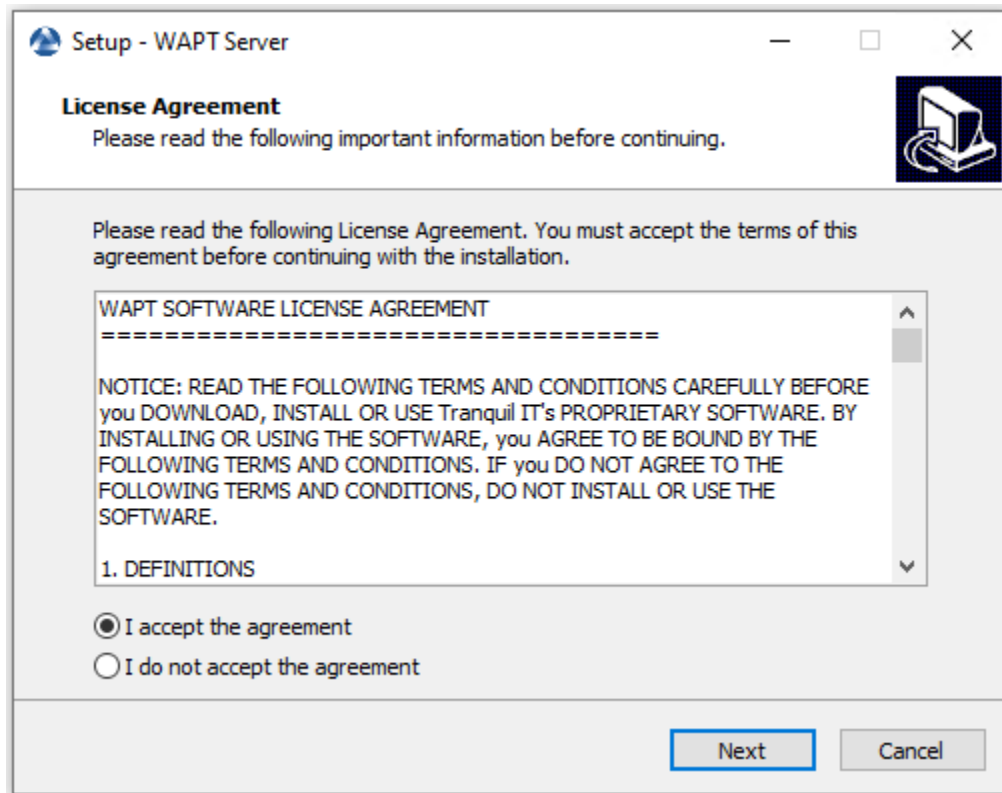


FIG. 2 – Acceptation des conditions de la licence WAPT

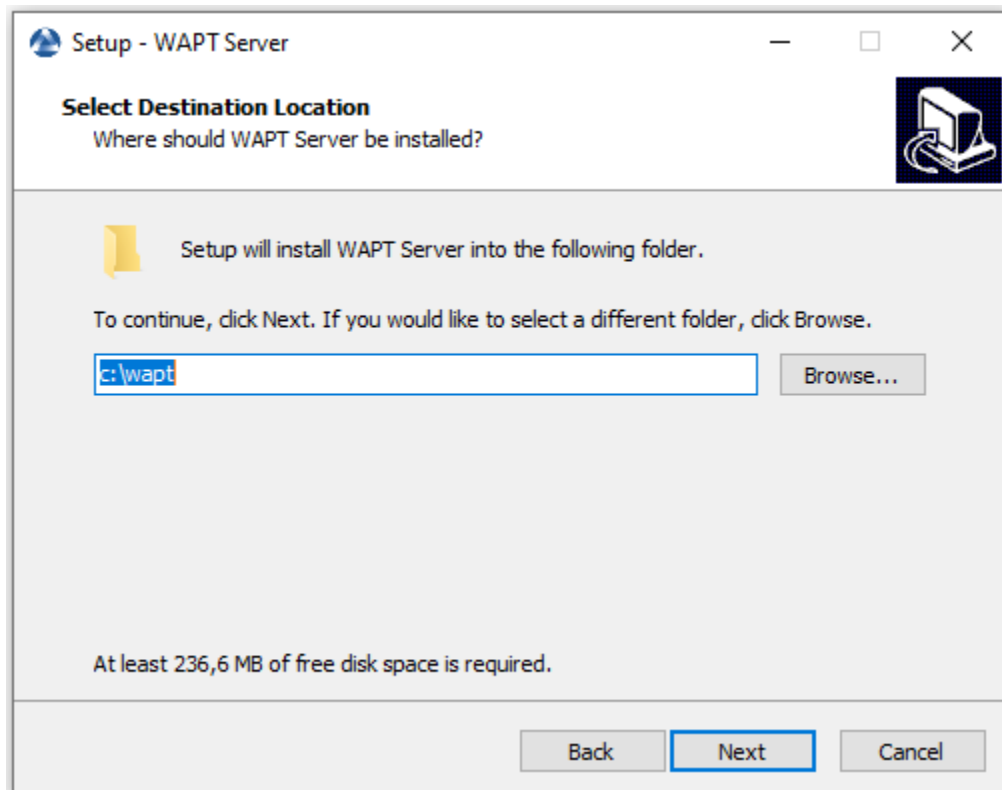


FIG. 3 – Choix du dossier de destination WAPT



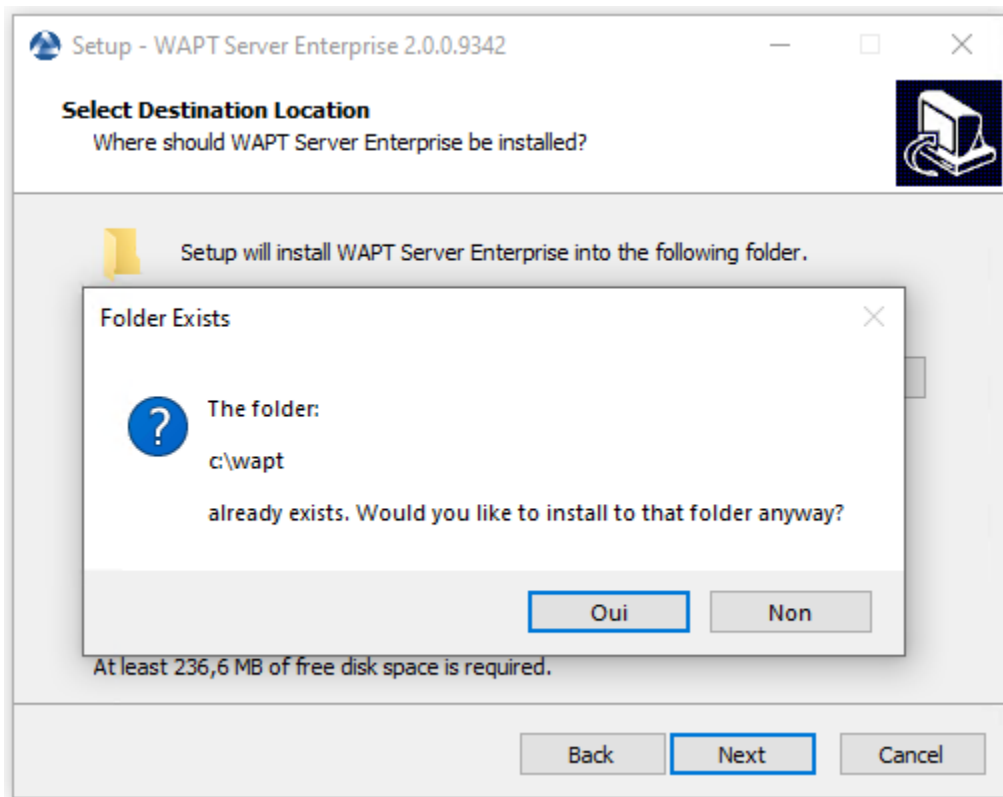
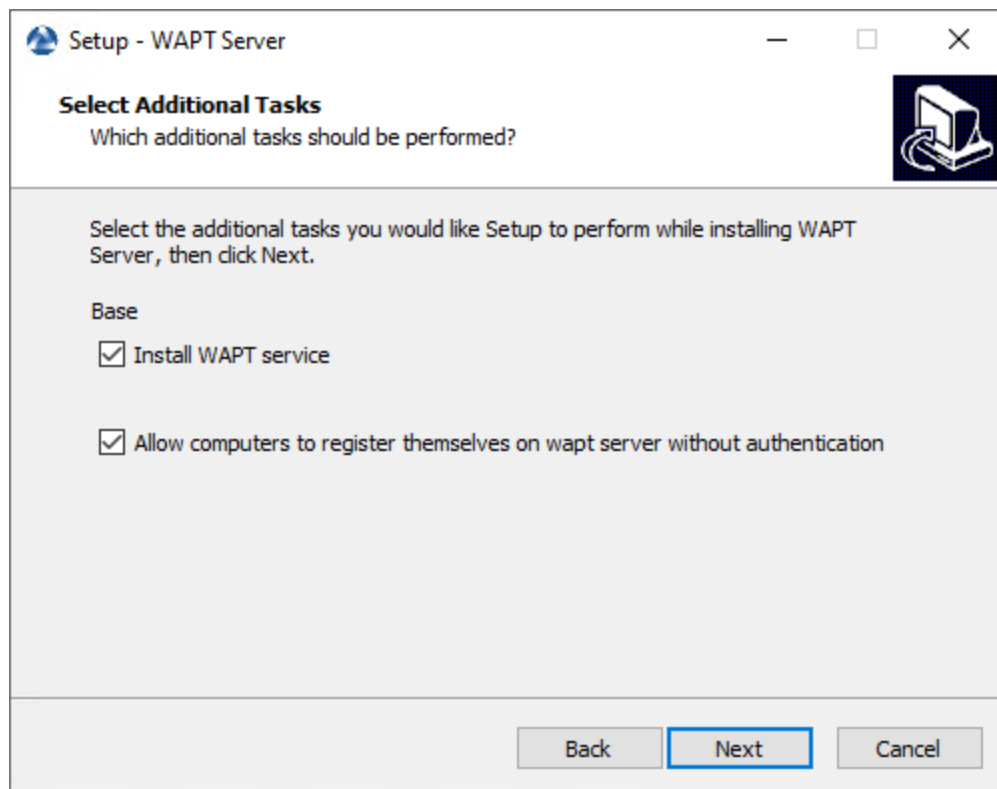


FIG. 4 – Message de l'ancien dossier Dossier de destination WAPT



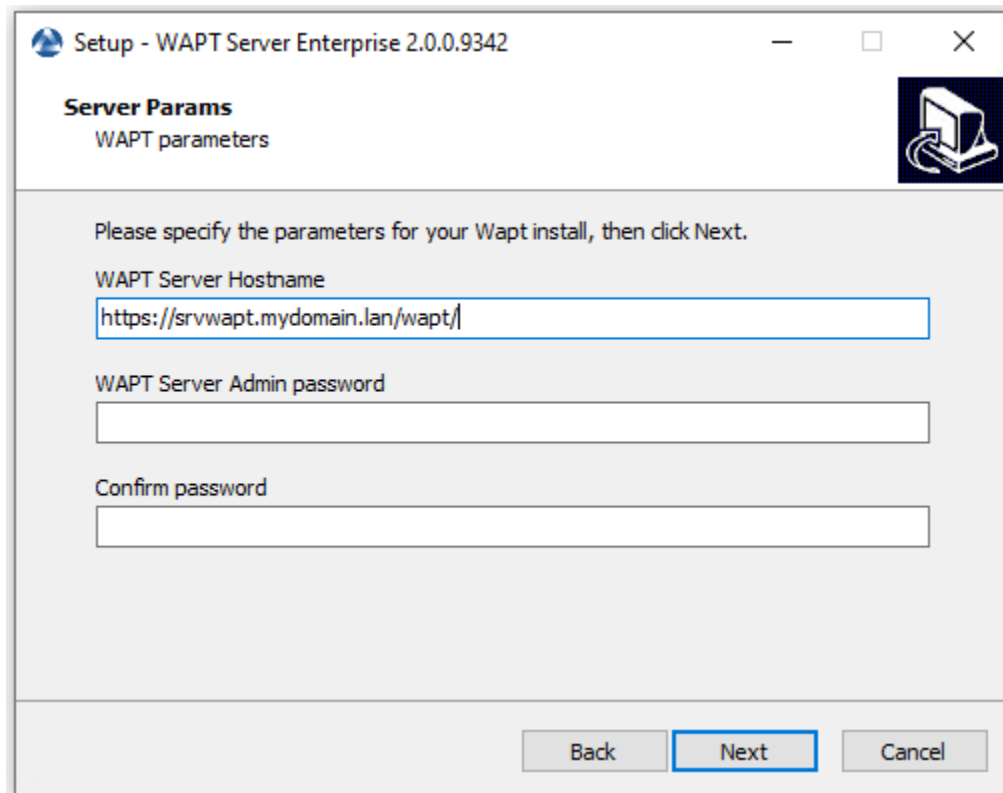


FIG. 5 – Modification du mot de passe du serveur WAPT

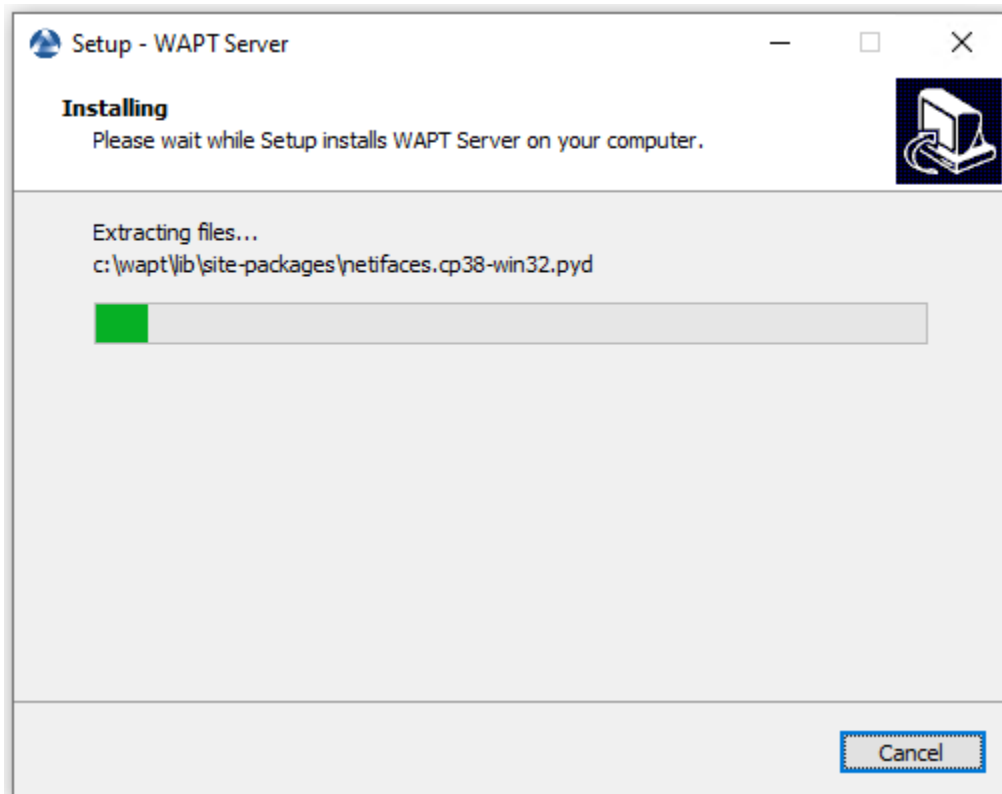
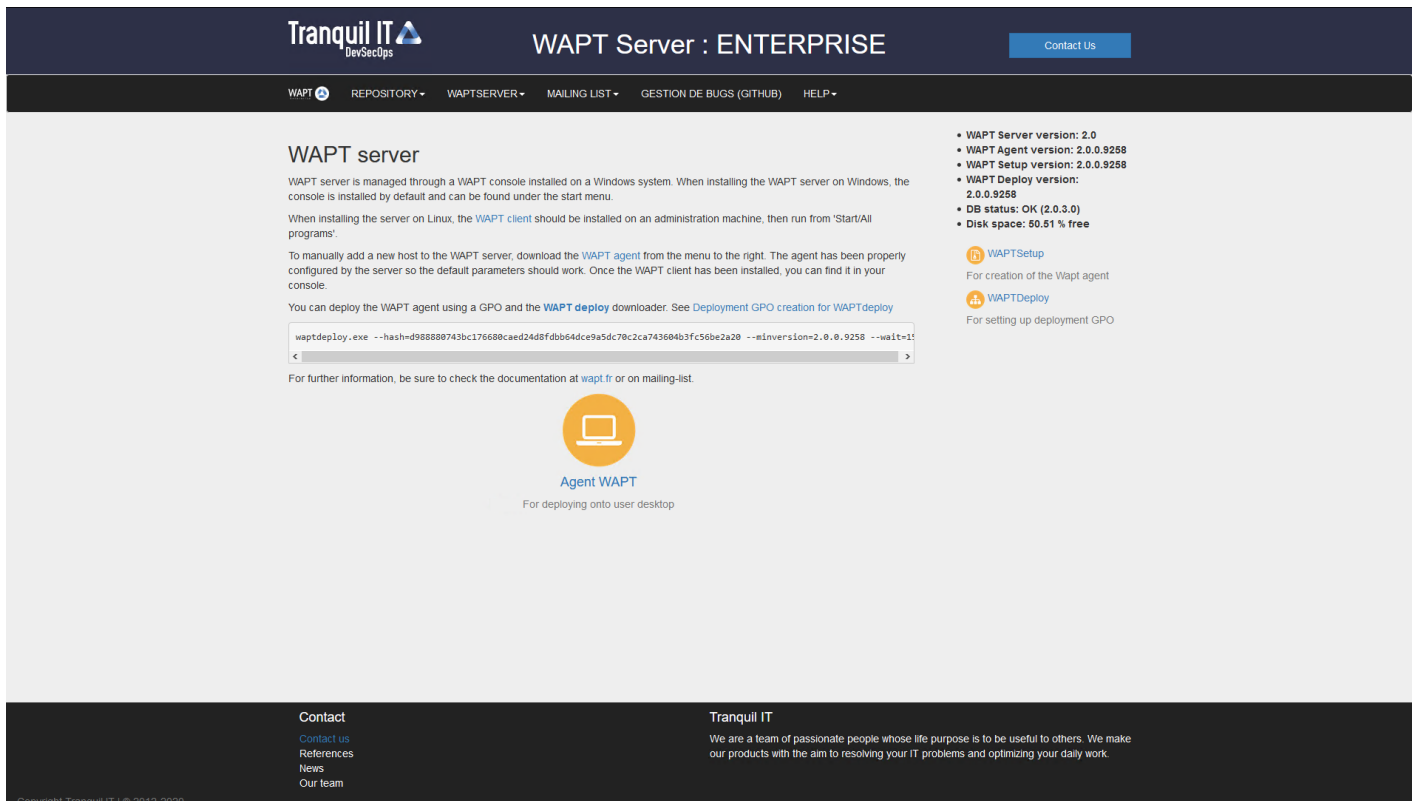


FIG. 6 – Installation du serveur WAPT en cours





---

### Mise à jour de WAPT de 1.8.2 à 2.1

---

---

**Indication :** Il n'est pas nécessaire de mettre à jour de la version 1.8.2 vers la 2.0 pour mettre à jour vers la 2.1.

---

Pour cette mise à jour, suivez *Mettre à jour WAPT de 1.8.2 vers 2.0* mais changez 2.0 en 2.1 sur les liens des dépôts.

**Avertissement :** Sur WAPT 2.1 *l'activation de la licence* est modifiée. Il est nécessaire d'exécuter le script de post-configuration.





---

## Mise à jour de WAPT de 1.8.2 à 2.0

---

---

**Note :** Avant de procéder à la mise à niveau, lisez les *exigences d'installation*.

---

**Attention :** Les ports 80 et 443 sont utilisés par le serveur WAPT et doivent être disponibles.

### 27.1 Debian

---

**Note :** Si vous êtes sous Debian 9 Stretch, vous devez d'abord mettre à niveau vers Debian 10 Buster avant de mettre à niveau vers WAPT 2.X. **Le serveur WAPT 2.X n'est pas disponible pour Debian 9.**

---

— Tout d'abord, mettez à jour la distribution.

```
apt update && apt upgrade -y
```

— Ajouter ou mettre à jour le dépôt de paquets Debian, importer la clé GPG du dépôt et installer les paquets du serveur WAPT.

### 27.1.1 De Community à Discovery

---

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>

---

### 27.1.2 Enterprise

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

---

— Installez `apt-transport-https` pour l'utilisation de https.

```
apt install apt-transport-https lsb-release gnupg
```

— Récupération de la clé `.gpg` et ajout dans le dépôt Tranquil IT.

```
wget -O - https://wapt.tranquil.it/debian/tiswapt-pub.gpg | apt-key add -  
echo "deb https://srvwapt-pro.tranquil.it/entreprise/debian/wapt-2.0/ $(lsb_release -c -s) main" > /  
↪etc/apt/sources.list.d/wapt.list
```

— S'il n'existe pas, créez `wapt.conf` dans `/etc/apt/auth.conf.d` pour stocker vos informations de connexion.

---

**Indication :** Remplacez `user` et `password` pour accéder au dépôt WAPT Enterprise, ce dernier est fourni pour notre département de vente.

---

```
cat > /etc/apt/auth.conf.d/wapt.conf <<EOF  
machine srvwapt-pro.tranquil.it  
login user  
password password  
EOF
```

— Appliquer les bons droits, mettre à jour le dépôt et installer les paquets.

```
chmod 600 /etc/apt/auth.conf.d/wapt.conf  
apt update  
apt install tis-waptserver tis-waptsetup
```

### 27.1.3 Lancer l'étape de post-configuration

- Lancer l'étape de post-configuration.

**Attention :** Pour que le post-configuration fonctionne correctement, vous devez d'abord avoir configuré correctement le *hostname* du serveur WAPT. Pour vérifier, utilisez la commande **echo \$(hostname)** qui doit vous renvoyer l'adresse DNS qui sera utilisé par les agents WAPT sur les ordinateurs clients.

**Avertissement :** Le script de post-configuration réécrit la configuration de nginx.

Si vous utilisez une *configuration spéciale*, sauvegardez votre fichier `wapt.conf` avec la commande

```
cp /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-available/wapt.conf.old
```

Il sera nécessaire d'écraser la configuration après la post-configuration avec la commande :

```
cp /etc/nginx/sites-available/wapt.conf.old /etc/nginx/sites-available/wapt.conf
```

**Indication :** Ce script de post-configuration doit être lancé en tant que **root**.

- Lancer le script.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

- Cliquez sur *Terminer* pour fermer la fenêtre.

```
do you want to launch post configuration tool?
```

```
< yes >          < no >
```

- Choisissez un mot de passe (si ce n'est pas déjà défini) pour le compte *SuperAdmin* du serveur WAPT (la longueur minimale est de 10 caractères).

```
Please enter the wapt server password (min. 10 characters)
```

```
*****
```

```
< OK >          < Cancel >
```

- Ne pas changer le mot de passe.

```
Please enter the server password again:
```

```
*****
```

```
< OK >          < Cancel >
```

- Choisissez le mode d'authentification pour l'enregistrement initial des agents WAPT :

- Le choix #1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
- Le choix #2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
- Le choix #3 n'active pas le mécanisme d'authentification kerberos pour l'enregistrement initial des machines équipées de WAPT. Le serveur WAPT demandera un login et un mot de passe pour chaque machine s'enregistrant auprès de lui.

```
WaptAgent Authentication type?
-----
(x) 1 Allow unauthenticated registration
( ) 2 Enable kerberos authentication required for machines registration.
    Registration will ask for password if kerberos not available
( ) 3 Disable kerberos but registration require strong authentication
-----
                < OK >          < Cancel >
```

— Sélectionnez *OK* pour démarrer le serveur WAPT.

```
Press OK to start waptserver
                < OK >
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
Do you want to configure nginx?
                < Yes >      < No >
```

— Renseignez le nom *FQDN* du serveur WAPT.

```
FQDN for the WAPT server (eg. wapt.example.com)
-----
wapt.mydomain.lan
-----
                < OK >          < Cancel >
```

— Sélectionnez *OK* et un certificat auto-signé sera généré, cette étape peut prendre un certain temps.

```
Generating DH parameters, 2048 bit long safe prime, generator 2 This is going to take a long time
.....+.....+....
```

Nginx est désormais configuré, sélectionnez *OK* pour redémarrer **Nginx** :

```
The Nginx config is done.
We need to restart Nginx?
                < OK >
```

Lancer l'étape de post-configuration.

```
Postconfiguration completed.
Please connect to https://wapt.mydomain.lan/ to access the server.
```

< OK >

Puis exécutez le script de post-configuration :

| Options       | Description                                                                                          |
|---------------|------------------------------------------------------------------------------------------------------|
| --force-https | Configurer <b>Nginx</b> de sorte que <i>le port 80 soit redirigé de façon permanente vers le 443</i> |

— Redémarrez le serveur WAPT.

```
systemctl restart waptserver nginx
```

— Assurez-vous que le propriétaire du fichier est correct.

```
chown wapt:www-data -R /var/www/wapt*
```

— Re-signez tous vos paquets WAPT.  
 — *En graphique*, ou  
 — *En ligne de commande*.

## 27.2 Ubuntu

— Tout d’abord, mettez à jour la distribution.

```
apt update && apt upgrade -y
```

— Ajouter ou mettre à jour le dépôt de paquets Debian, importer la clé GPG du dépôt et installer les paquets du serveur WAPT.

### 27.2.1 De Community à Discovery

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>

### 27.2.2 Enterprise

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l’édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

— Installez `apt-transport-https` pour l’utilisation de https.

```
apt install apt-transport-https lsb-release gnupg
```

— Récupération de la clé .gpg et ajout dans le dépôt Tranquil IT.

```
wget -O - https://wapt.tranquil.it/debian/tiswapt-pub.gpg | apt-key add -  
echo "deb https://srvwapt-pro.tranquil.it/entreprise/ubuntu/wapt-2.0/ $(lsb_release -c -s) main" > /  
etc/apt/sources.list.d/wapt.list
```

— S'il n'existe pas, créez wapt.conf dans /etc/apt/auth.conf.d pour stocker vos informations de connexion.

---

**Indication :** Remplacez **user** et **password** pour accéder au dépôt WAPT Enterprise, ce dernier est fourni pour notre département de vente.

---

```
cat > /etc/apt/auth.conf.d/wapt.conf <<EOF  
machine srvwapt-pro.tranquil.it  
login user  
password password  
EOF
```

— Appliquer les bons droits, mettre à jour le dépôt et installer les paquets.

```
chmod 600 /etc/apt/auth.conf.d/wapt.conf  
apt update  
apt install tis-waptserver tis-waptsetup
```

### 27.2.3 Lancer l'étape de post-configuration

— Lancer l'étape de post-configuration.

**Attention :** Pour que le post-configuration fonctionne correctement, vous devez d'abord avoir configuré correctement le *hostname* du serveur WAPT. Pour vérifier, utilisez la commande **echo \$(hostname)** qui doit vous renvoyer l'adresse DNS qui sera utilisé par les agents WAPT sur les ordinateurs clients.

**Avertissement :** Le script de post-configuration réécrit la configuration de nginx.

Si vous utilisez une *configuration spéciale*, sauvegardez votre fichier wapt.conf avec la commande

```
cp /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-available/wapt.conf.old
```

Il sera nécessaire d'écraser la configuration après la post-configuration avec la commande :

```
cp /etc/nginx/sites-available/wapt.conf.old /etc/nginx/sites-available/wapt.conf
```

---

**Indication :** Ce script de post-configuration doit être lancé en tant que **root**.

---

— Lancer le script.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
do you want to launch post configuration tool?
```

```
< yes >          < no >
```

— Choisissez un mot de passe (si ce n'est pas déjà défini) pour le compte *SuperAdmin* du serveur WAPT (la longueur minimale est de 10 caractères).

```
Please enter the wapt server password (min. 10 characters)
```

```
*****
```

```
< OK >          < Cancel >
```

— Ne pas changer le mot de passe.

```
Please enter the server password again:
```

```
*****
```

```
< OK >          < Cancel >
```

— Choisissez le mode d'authentification pour l'enregistrement initial des agents WAPT :

- Le choix #1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
- Le choix #2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
- Le choix #3 n'active pas le mécanisme d'authentification kerberos pour l'enregistrement initial des machines équipées de WAPT. Le serveur WAPT demandera un login et un mot de passe pour chaque machine s'enregistrant auprès de lui.

```
WaptAgent Authentication type?
```

- ```
-----
(x) 1 Allow unauthenticated registration
( ) 2 Enable kerberos authentication required for machines registration.
    Registration will ask for password if kerberos not available
( ) 3 Disable kerberos but registration require strong authentication
-----
```

```
< OK >          < Cancel >
```

— Sélectionnez *OK* pour démarrer le serveur WAPT.

```
Press OK to start waptserver
```

```
< OK >
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
Do you want to configure nginx?
```

```
< Yes >      < No >
```

— Renseignez le nom *FQDN* du serveur WAPT.

```
FQDN for the WAPT server (eg. wapt.example.com)
```

```
-----  
wapt.mydomain.lan  
-----
```

```
< OK >      < Cancel >
```

— Sélectionnez *OK* et un certificat auto-signé sera généré, cette étape peut prendre un certain temps.

```
Generating DH parameters, 2048 bit long safe prime, generator 2 This is going to take a long time  
.....+.....+....
```

Nginx est désormais configuré, sélectionnez *OK* pour redémarrer **Nginx** :

```
The Nginx config is done.  
We need to restart Nginx?
```

```
< OK >
```

Lancer l'étape de post-configuration.

```
Postconfiguration completed.  
Please connect to https://wapt.mydomain.lan/ to access the server.
```

```
< OK >
```

Puis exécutez le script de post-configuration :

Options	Description
<code>--force-https</code>	Configurer <b>Nginx</b> de sorte que <i>le port 80 soit redirigé de façon permanente vers le 443</i>

— Redémarrez le serveur WAPT.

```
systemctl restart waptserver nginx
```

— Assurez-vous que le propriétaire du fichier est correct.

```
chown wapt:www-data -R /var/www/wapt*
```

— Re-signez tous vos paquets WAPT.

- *En graphique*, ou
- *En ligne de commande*.



## 27.3 CentOS / RedHat

— Tout d’abord, mettez à jour la distribution.

```
yum update
```

— Ajouter ou mettre à jour le dépôt de paquets Debian, importer la clé GPG du dépôt et installer les paquets du serveur WAPT.

### 27.3.1 CentOS7

#### De Community à Discovery

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l’édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

---

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>.

---

#### Enterprise

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l’édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

Pour accéder au site de téléchargement WAPT Enterprise, vous devez utiliser le nom d’utilisateur et le mot de passe fournis par notre service commercial.

---

— Ajout du dépôt Tranquil IT.

```
cat > /etc/yum.repos.d/wapt.repo <<EOF
[wapt]
name=WAPT Server Repo
baseurl=https://srvwapt-pro.tranquil.it/entreprise/centos7/wapt-2.0/
enabled=1
gpgcheck=1
EOF
```

### Installez tous les paquets nécessaires

— Récupération de la clé .gpg.

```
wget -q -O /tmp/tranquil_it.gpg "https://wapt.tranquil.it/centos7/RPM-GPG-KEY-TISWAPT-7"; rpm --  
→import /tmp/tranquil_it.gpg
```

— Installez tous les paquets nécessaires

```
yum install epel-release -y  
yum install postgresql96-server postgresql96-contrib tis-waptserver tis-waptsetup cabextract -y
```

## 27.3.2 CentOS8

### De Community à Discovery

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

---

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>.

---

### Enterprise

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

Pour accéder au site de téléchargement WAPT Enterprise, vous devez utiliser le nom d'utilisateur et le mot de passe fournis par notre service commercial.

---

— Ajout du dépôt Tranquil iT

```
cat > /etc/yum.repos.d/wapt.repo <<EOF  
[wapt]  
name=WAPT Server Repo  
baseurl=https://srvwapt-pro.tranquil.it/entreprise/centos8/wapt-2.0/  
enabled=1  
gpgcheck=1  
EOF
```

## Installez tous les paquets nécessaires

— Récupération de la clé .gpg.

```
wget -q -O /tmp/tranquil_it.gpg "https://wapt.tranquil.it/centos7/RPM-GPG-KEY-TISWAPT-7"; rpm --
↳import /tmp/tranquil_it.gpg
```

— Installez tous les paquets nécessaires

```
yum install epel-release -y
yum install postgresql96-server postgresql96-contrib tis-waptserver tis-waptsetup cabextract -y
```

### 27.3.3 Lancer l'étape de post-configuration

**Attention :** Pour que le post-configuration fonctionne correctement, vous devez d'abord avoir configuré correctement le *hostname* du serveur WAPT. Pour vérifier, utilisez la commande **echo \$(hostname)** qui doit vous renvoyer l'adresse DNS qui sera utilisé par les agents WAPT sur les ordinateurs clients.

**Avertissement :** Le script de post-configuration réécrit la configuration de nginx.

Si vous utilisez une *configuration spéciale*, sauvegardez votre fichier `wapt.conf` avec la commande

```
cp /etc/nginx/sites-available/wapt.conf /etc/nginx/sites-available/wapt.conf.old
```

Il sera nécessaire d'écraser la configuration après la post-configuration avec la commande :

```
cp /etc/nginx/sites-available/wapt.conf.old /etc/nginx/sites-available/wapt.conf
```

**Indication :** Ce script de post-configuration doit être lancé en tant que **root**.

— Lancer le script.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

— Cliquez sur *Terminer* pour fermer la fenêtre.

```
do you want to launch post configuration tool?
```

```
< yes >          < no >
```

— Choisissez un mot de passe (si ce n'est pas déjà défini) pour le compte *SuperAdmin* du serveur WAPT (la longueur minimale est de 10 caractères).

```
Please enter the wapt server password (min. 10 characters)
```

```
*****
```

(suite sur la page suivante)



```
Generating DH parameters, 2048 bit long safe prime, generator 2 This is going to take a long time
.....+.....+...
```

Nginx est désormais configuré, sélectionnez *OK* pour redémarrer **Nginx** :

```
The Nginx config is done.
We need to restart Nginx?
```

< OK >

Lancer l'étape de post-configuration.

```
Postconfiguration completed.
Please connect to https://wapt.mydomain.lan/ to access the server.
```

< OK >

Puis exécutez le script de post-configuration :

Options	Description
<code>--force-https</code>	Configurer <b>Nginx</b> de sorte que <i>le port 80 soit redirigé de façon permanente vers le 443</i>

— Redémarrez le serveur WAPT.

```
systemctl restart waptserver nginx
```

— Assurez-vous que le propriétaire du fichier est correct.

```
chown wapt:www-data -R /var/www/wapt*
```

— Re-signez tous vos paquets WAPT.

— *En graphique*, ou

— *En ligne de commande*.

## 27.4 Windows

### 27.4.1 De Community à Discovery

---

**Note** : Non disponible à la date du 2024-01-09.

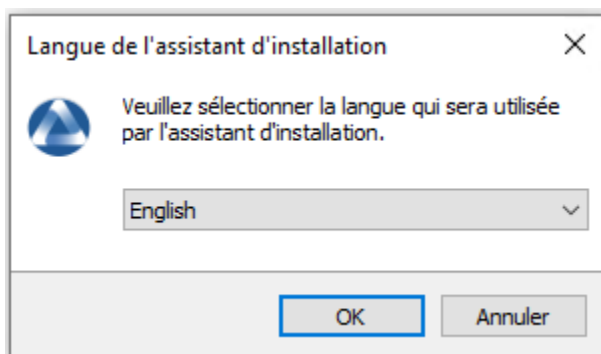
---

## 27.4.2 Enterprise

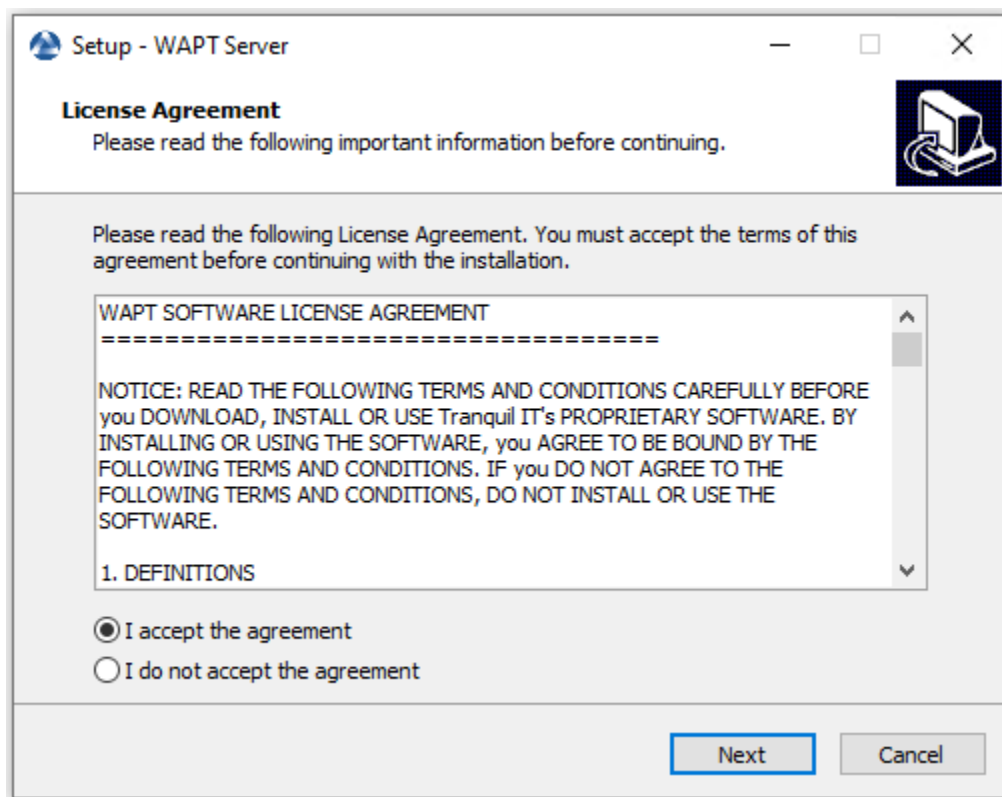
**Indication :** Pour accéder aux ressources de WAPT Enterprise, vous devez utiliser le nom d'utilisateur et le mot de passe fournis par notre service commercial.

---

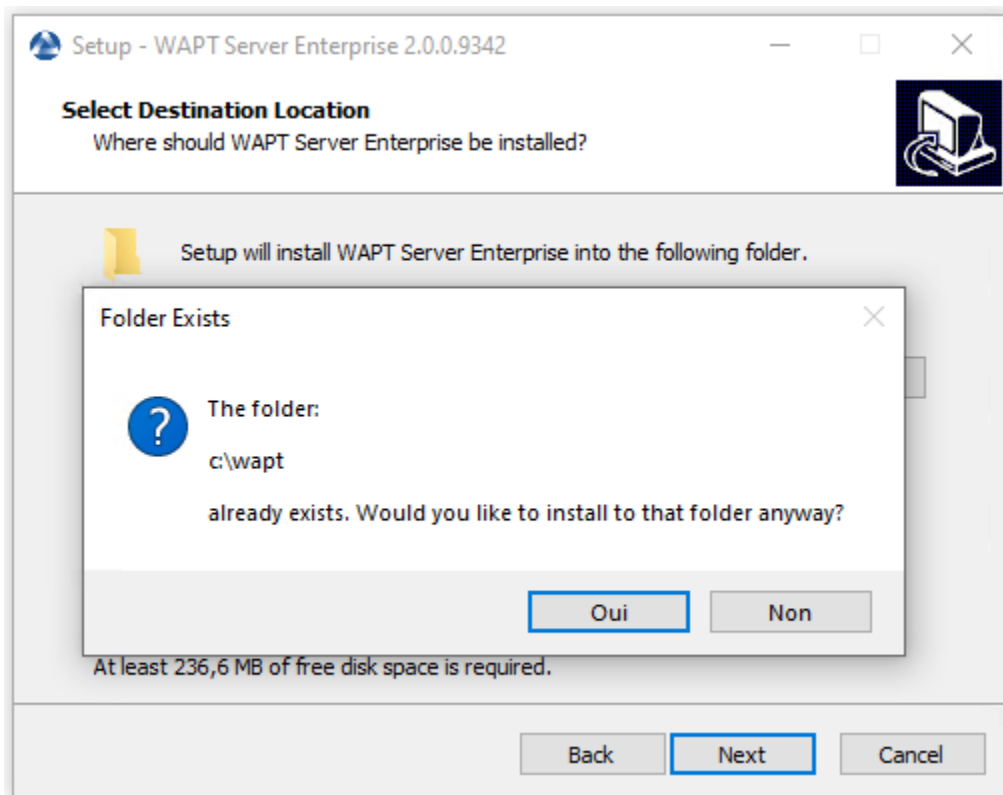
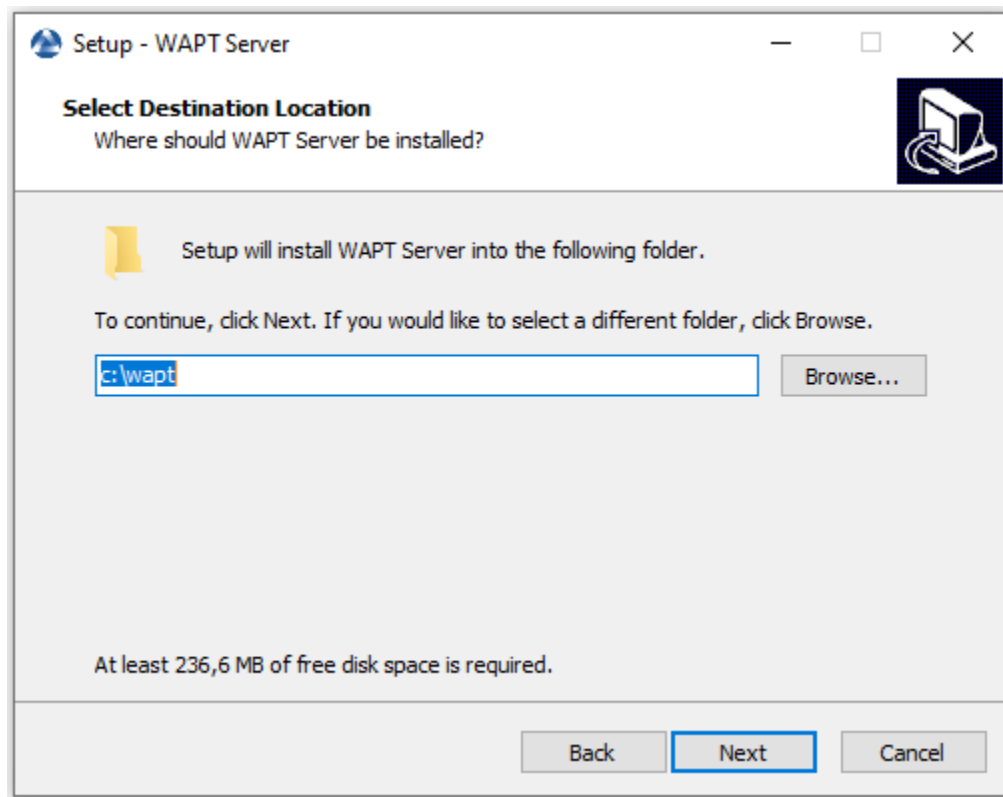
- Téléchargez et exécutez waptserversetup.exe.
- Choisissez la langue d'installation.



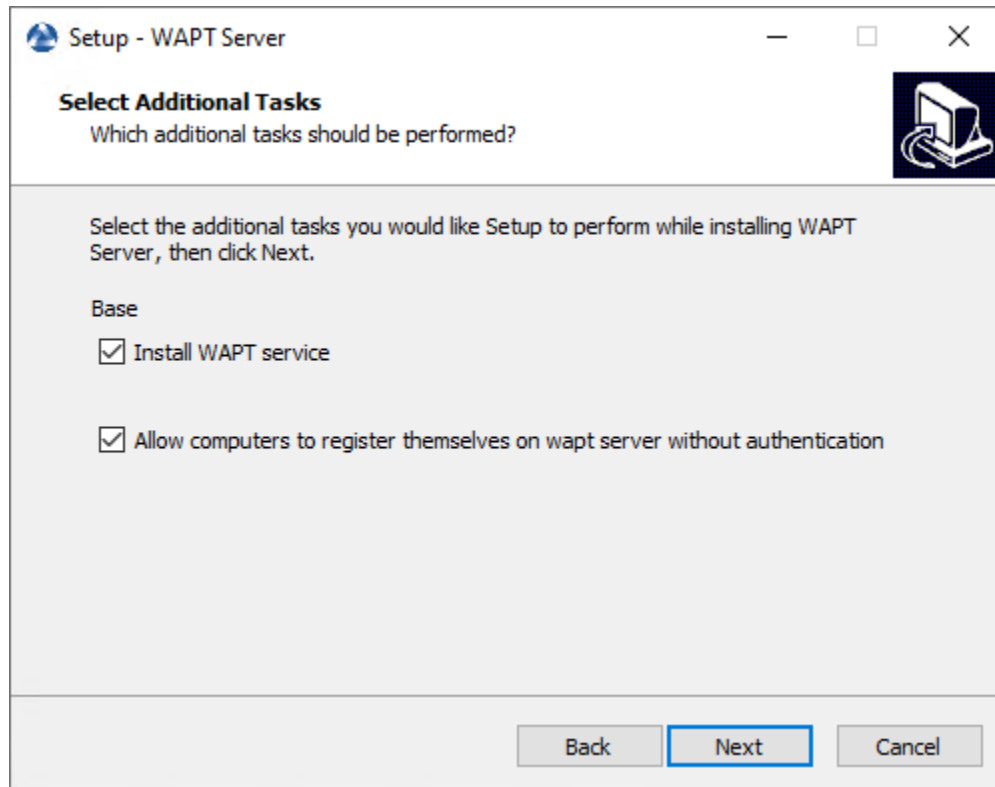
- Acceptez la licence publique GNU et cliquez sur *Suivant* pour passer à l'étape suivante.



- Choisissez le répertoire d'installation (laissez la valeur par défaut) et cliquez sur *Suivant* pour passer à l'étape suivante.
- Choisissez le répertoire d'installation (laissez la valeur par défaut) et cliquez sur *Suivant* pour passer à l'étape suivante.



- Sélectionnez une tâche supplémentaire si nécessaire.



- Modifiez le mot de passe du serveur WAPT si nécessaire, puis appuyez sur *Suivant*.
- Cliquez sur le bouton *Installer* pour lancer l'installation, attendez que l'installation soit terminée.
- Cliquez sur *Terminer* pour fermer la fenêtre.

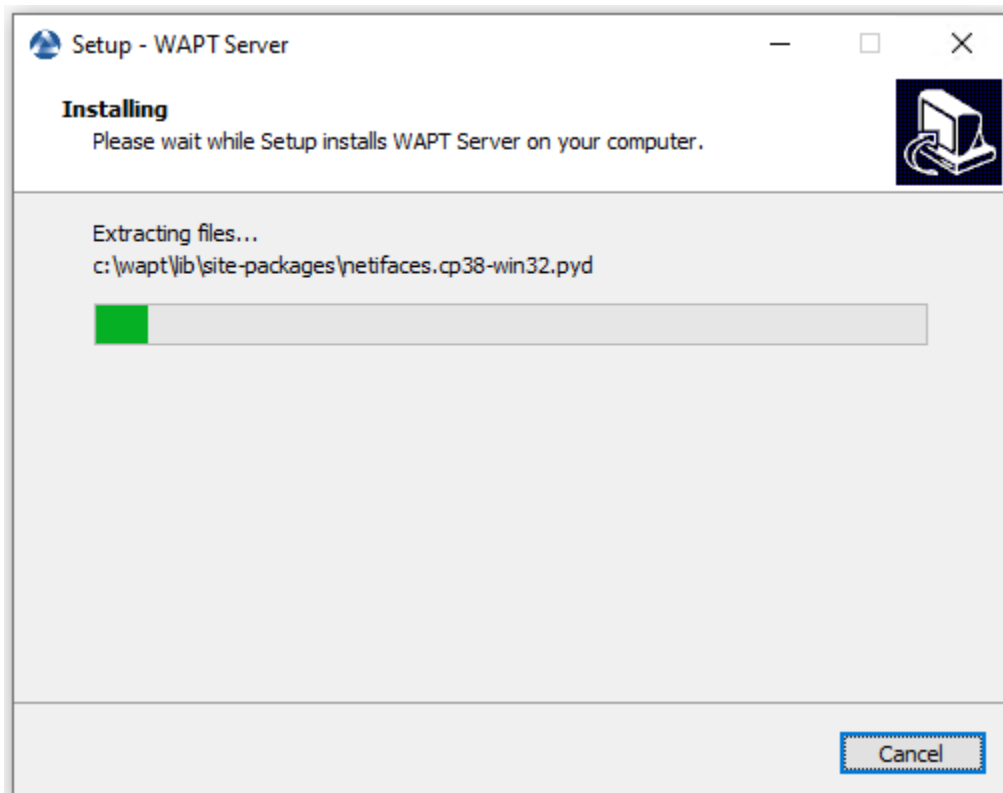
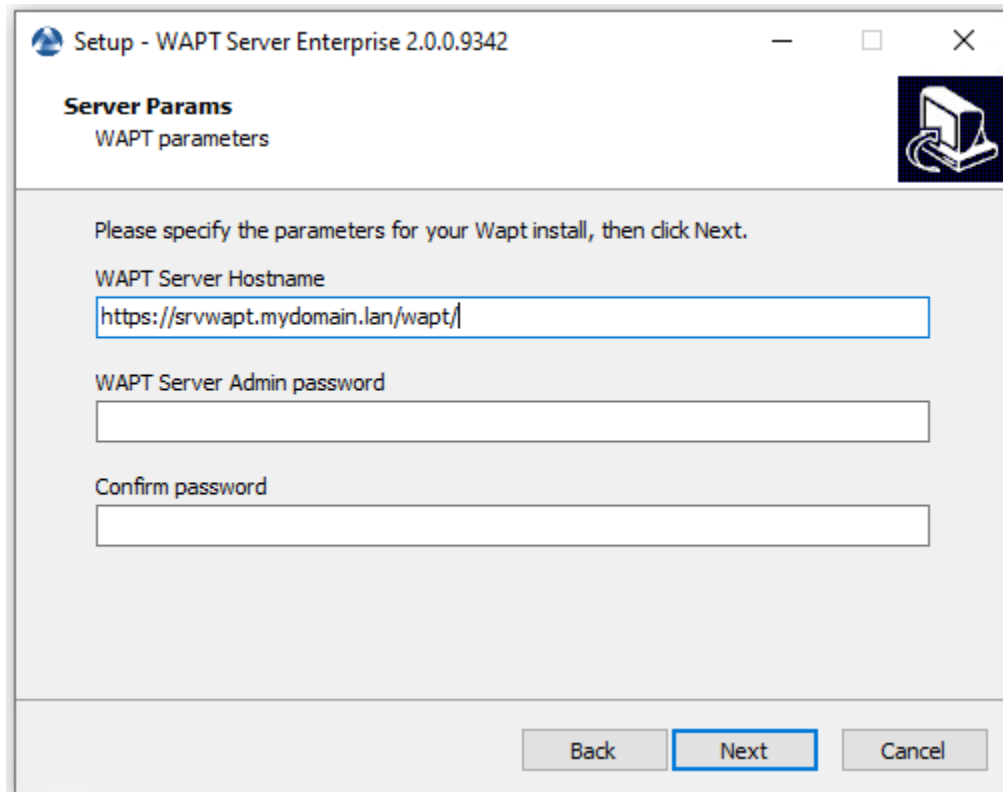
Le serveur WAPT sur votre serveur ou station de travail Windows est prêt.

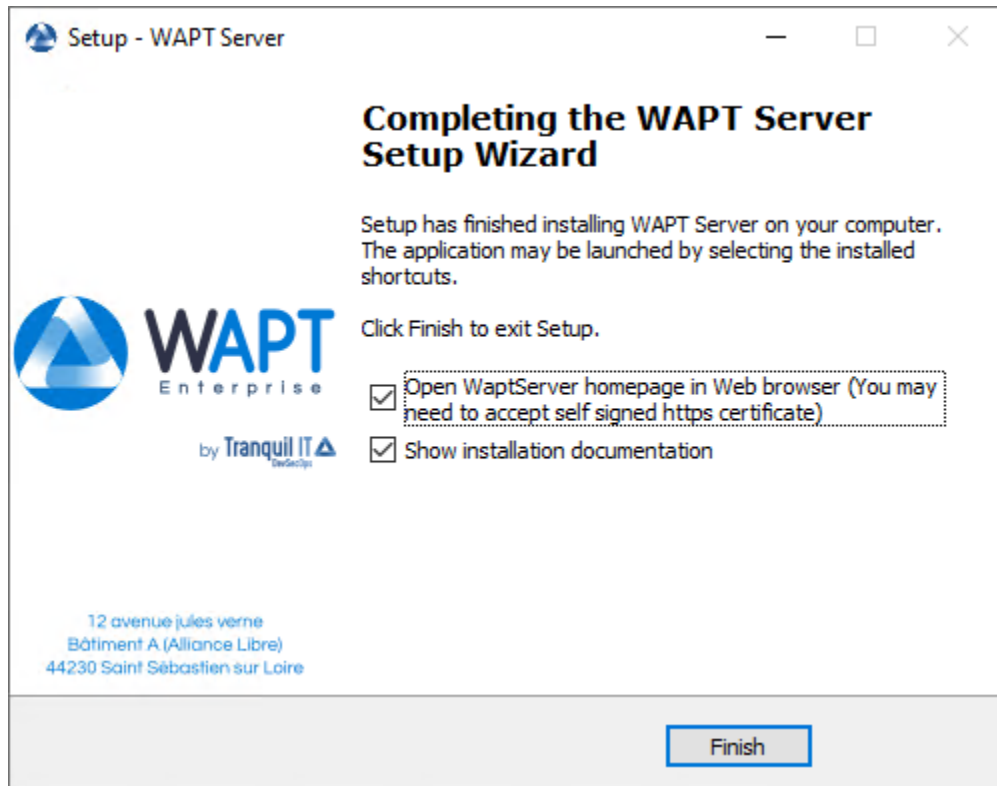
**Attention : NE PAS** utiliser la console WAPT sur le serveur WAPT. **N'installez PAS** et n'exécutez pas vos outils de développement de paquets WAPT sur le serveur WAPT.

Votre serveur est maintenant prêt. Vous pouvez maintenant consulter la documentation sur *Installation de la console de gestion WAPT*.

- Re-signez tous vos paquets WAPT.
  - *En graphique*







Tranquil IT DevSecOps
WAPT Server : ENTERPRISE [Contact Us](#)

---

WAPT
REPOSITORY
WAPTSERVER
MAILING LIST
GESTION DE BUGS (GITHUB)
HELP

### WAPT server

WAPT server is managed through a WAPT console installed on a Windows system. When installing the WAPT server on Windows, the console is installed by default and can be found under the start menu.

When installing the server on Linux, the WAPT client should be installed on an administration machine, then run from 'Start/All programs'.

To manually add a new host to the WAPT server, download the [WAPT agent](#) from the menu to the right. The agent has been properly configured by the server so the default parameters should work. Once the WAPT client has been installed, you can find it in your console.

You can deploy the WAPT agent using a GPO and the [WAPT deploy](#) downloader. See [Deployment GPO creation for WAPTdeploy](#)

```
waptdeploy.exe --hash=d98889743bc176680caed24dfdbb64dce9a5dc78c2ca743604b3fc56be2a28 --mInversion=2.0.0.9258 --wait=1
```

For further information, be sure to check the documentation at [wapt.fr](#) or on mailing-list.

**Agent WAPT**

For deploying onto user desktop

- WAPT Server version: 2.0
- WAPT Agent version: 2.0.0.9258
- WAPT Setup version: 2.0.0.9258
- WAPT Deploy version: 2.0.0.9258
- DB status: OK (2.0.3.0)
- Disk space: 50.51 % free

**WAPT Setup**  
For creation of the Wapt agent

**WAPT Deploy**  
For setting up deployment GPO

**Contact**

[Contact us](#)

[References](#)

[News](#)

[Our team](#)

**Tranquil IT**

We are a team of passionate people whose life purpose is to be useful to others. We make our products with the aim of resolving your IT problems and optimizing your daily work

Copyright Tranquil IT | © 2012-2020

---

Mettre à jour de 1.6/1.7 vers 1.8

---

## 28.1 Debian

**Attention :** Les ports 80 et 443 sont utilisés par le serveur WAPT et doivent être disponibles.

- Tout d’abord, mettez à jour la distribution sous-jacente de Debian.

```
apt update && apt upgrade -y && apt dist-upgrade
```

- Ajouter le dépôt de paquets pour les paquets Debian, importer la clé GPG du dépôt et installer les paquets du serveur WAPT.

### 28.1.1 Enterprise

**Indication :** Pour accéder au site de téléchargement WAPT Enterprise, vous devez utiliser le nom d’utilisateur et le mot de passe fournis par notre service commercial.

Remplacez **user** et **password** dans le paramètre **deb** pour accéder au dépôt WAPT Enterprise.

---

```
apt install apt-transport-https lsb-release
wget -O - https://wapt.tranquil.it/debian/tiswapt-pub.gpg | apt-key add -
echo "deb https://user:password@srvwapt-pro.tranquil.it/enterprise/debian/wapt-1.8/ $(lsb_release
↵-c -s) main" > /etc/apt/sources.list.d/wapt.list
apt update
apt install tis-waptserver tis-waptsetup
```

## 28.1.2 Community

```
apt install apt-transport-https lsb-release
wget -O - https://wapt.tranquil.it/debian/tiswapt-pub.gpg | apt-key add -
echo "deb https://wapt.tranquil.it/debian/wapt-1.8/ $(lsb_release -c -s) main" > /etc/apt/
↳sources.list.d/wapt.list
apt update
apt install tis-waptserver tis-waptsetup
```

## 28.1.3 Lancer l'étape de post-configuration

- Lancer l'étape de post-configuration.

---

### Note :

- Nous vous conseillons de lancer les étapes de post-configuration après chaque mise à niveau du serveur afin que le serveur utilise le dernier format de configuration.
  - Il n'est pas nécessaire de réinitialiser un mot de passe pour la console WAPT pendant l'étape de post-configuration.
  - Si vous avez personnalisé la configuration de **Nginx**, ne répondez pas à *Yes* lorsque la post-configuration vous demande de configurer **Nginx**.
- 

### Attention :

- Avec la post-configuration de WAPT 1.8, les paquets WUA de WAPT seront déplacés de leur emplacement de stockage actuel vers le dossier racine de waptwua (`/var/www/waptwua`).
- Si la réplication du dépôt a été définie, tous les paquets KB/CAB seront resynchronisés sur les référentiels distants.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

Le mot de passe demandé à l'étape 4 est utilisé pour accéder à la console WAPT.

- Démarrez le serveur WAPT.

```
systemctl restart waptserver
```

- Mettez à niveau la console WAPT en suivant le même ensemble d'étapes que *installation de la console WAPT*.
- Puis *créer l'agent WAPT*.  
Cependant, il faudra conserver le même préfixe et surtout ne rien changer concernant le couple clé privée / clé publique ! Cela va générer un paquet **waptupgrade** dans le dépôt privé.

---

**Note :** Il y a deux manières de déployer la mise à jour :

- En utilisant un GPO et **waptdeploy**.
  - Utiliser un paquet **waptupgrade** et le déployer en utilisant WAPT.
- 

- Mettre à jour les agents WAPT.  
La procédure à suivre pour la mise à jour des clients est la même que pour la première installation. Télécharger et installer la dernière version de l'agent WAPT qui a été mis à jour : <https://wapt.mydomain.lan/wapt/waptagent.exe>.  
Comme mentionné ci-dessus, cette procédure peut s'automatiser avec une GPO ou un paquet de mise à jour **waptupgrade**.
- *Mise à jour mineure pour CentOS.*
- *Mise à jour mineure pour Windows.*

**Attention :**

- Debian Jessie est maintenant obsolète. **WAPT 1.8 ne fonctionnera pas avec les anciennes versions de Debian.**
- Envisagez de migrer votre installation WAPT existante vers Debian Buster ou CentOS7.

## 28.2 CentOS

- Tout d’abord, mettez à jour la distribution sous-jacente CentOS/ RedHat.

```
yum update
```

### 28.2.1 Enterprise

Modifier l’adresse du dépôt puis lancer la mise à niveau.

**Indication :** Pour accéder au site de téléchargement WAPT Enterprise, vous devez utiliser le nom d’utilisateur et le mot de passe fournis par notre service commercial.

Remplacez **user** et **password** dans le paramètre **baseurl** pour accéder au dépôt WAPT Enterprise.

```
cat > /etc/yum.repos.d/wapt.repo <<EOF
[wapt]
name=WAPT Enterprise Server Repo
baseurl=https://user:password@srvwapt-pro.tranquil.it/entreprise/centos7/wapt-1.8/
enabled=1
gpgcheck=1
EOF

wget -q -O /tmp/tranquil_it.gpg "https://wapt.tranquil.it/centos7/RPM-GPG-KEY-TISWAPT-7"; rpm --
→import /tmp/tranquil_it.gpg
yum install epel-release
yum install cabextract
yum install postgresql96-server postgresql96-contrib tis-waptserver tis-waptsetup
```

### 28.2.2 Community

- Modifier l’adresse du dépôt puis lancer la mise à niveau.

```
cat > /etc/yum.repos.d/wapt.repo <<EOF
[wapt]
name=WAPT Server Repo
baseurl=https://wapt.tranquil.it/centos7/wapt-1.8/
enabled=1
gpgcheck=1
EOF
```

(suite sur la page suivante)

(suite de la page précédente)

```
wget -q -O /tmp/tranquil_it.gpg "https://wapt.tranquil.it/centos7/RPM-GPG-KEY-TISWAPT-7"; rpm --
→import /tmp/tranquil_it.gpg
yum install postgresql96-server postgresql96-contrib tis-waptserver tis-waptsetup
```

## 28.2.3 Lancer l'étape de post-configuration

- Lancer l'étape de post-configuration.

### Note :

- Nous vous conseillons de lancer les étapes de post-configuration après chaque mise à niveau du serveur afin que le serveur utilise le dernier format de configuration.
- Il n'est pas nécessaire de réinitialiser un mot de passe pour la console WAPT pendant l'étape de post-configuration.
- Si vous avez personnalisé la configuration de **Nginx**, ne répondez pas à *Yes* lorsque la post-configuration vous demande de configurer **Nginx**.

### Attention :

- Avec la post-configuration de WAPT 1.8, les paquets WUA de WAPT seront déplacés de leur emplacement de stockage actuel vers le dossier racine de waptwua (/var/www/waptwua).
- Si la réplication du dépôt a été définie, tous les paquets KB/CAB seront resynchronisés sur les référentiels distants.

```
/opt/wapt/waptserver/scripts/postconf.sh
```

- Démarrez le serveur WAPT.

```
systemctl start waptserver
```

- Mettez à niveau la console WAPT en suivant le même ensemble d'étapes que *installation de la console WAPT*.
- Puis *créer l'agent WAPT*.  
Cependant, il faudra conserver le même préfixe et surtout ne rien changer concernant le couple clé privée / clé publique ! Cela va générer un paquet **waptupgrade** dans le dépôt privé.

**Note :** Il y a deux manières de déployer la mise à jour :

- En utilisant un GPO et **waptdeploy**.
- Utiliser un paquet **waptupgrade** et le déployer en utilisant WAPT.

- Mettre à jour les agents WAPT.

La procédure à suivre pour la mise à jour des clients est la même que pour la première installation.

Télécharger et installer la dernière version de l'agent WAPT qui a été mis à jour : <https://wapt.mydomain.lan/wapt/waptagent.exe>.

Comme mentionné ci-dessus, cette procédure peut s'automatiser avec une GPO ou un paquet de mise à jour **waptupgrade**.

## 28.3 Windows

**Attention :** En cas d'utilisation d'une GPO pour la mise à jour de l'agent WAPT, il faut exclure le serveur WAPT de l'OU où est appliqué la GPO.

**Note :** Le serveur WAPT peut s'installer sur Windows 7 64bit et Windows 10 64bit et Windows Server 2012/2012R2, 2016, 2019 et 2022 64bit.

- Sur la machine Windows hébergeant le serveur WAPT, téléchargez la dernière version du programme d'installation depuis le site web de Tranquil IT *WAPTServerSetup.exe* : *download* : <<https://wapt.tranquil.it/wapt/releases/latest/waptserversetup.exe>> et lancez-le en tant que *Administrateur local*.
- Installer la mise à jour.

**Note :** La procédure à suivre est la même que celle pour *installer un serveur WAPT sur Windows*.

**Attention :** Le préfixe ne doit pas changer et vous **NE DEVEZ PAS** régénérer de clé!

- Sur la station de travail que vous utilisez pour construire vos paquets, téléchargez manuellement WAPTSetup à partir de.
  - Discovery : *waptsetup.exe*.
  - Enterprise : *waptsetup.exe*.
- Puis *créer l'agent WAPT*.  
Cependant, il faudra conserver le même préfixe et surtout ne rien changer concernant le couple clé privée / clé publique ! Cela va générer un paquet **waptupgrade** dans le dépôt privé.

**Note :** Il y a deux manières de déployer la mise à jour :

- En utilisant un GPO et **waptdeploy**.
- Utiliser un paquet **waptupgrade** et le déployer en utilisant WAPT.

- Mettre à jour les agents WAPT.  
La procédure à suivre pour la mise à jour des clients est la même que pour la première installation. Téléchargez et installez la dernière version de l'agent WAPT en vous rendant sur <http://wapt.mydomain.lan/wapt/waptagent.exe> (**le lien est votre propre serveur WAPT**).  
Comme mentionné ci-dessus, cette procédure peut s'automatiser avec une GPO ou un paquet de mise à jour **waptupgrade**.





---

### Mettre à jour WAPT depuis des version antérieur à 1.6

---

Nous vous recommandons de repartir de zéro, que vous mettiez en place un serveur WAPT à jour à côté de l'ancienne version, et que vous utilisiez l'ancienne configuration ou une GPO du domaine à déployer sur les agents WAPT.



---

## Sauvegarder le serveur WAPT

---

Pour sauvegarder votre serveur, suivez cette procédure. Des sauvegardes régulières sont recommandées.

### 30.1 Linux

- Arrêter les services liés à WAPT sur le serveur.

```
systemctl stop waptasks
systemctl stop waptserver
systemctl stop nginx
```

- Sauvegarder ces répertoires en utilisant un outil de sauvegarde (ex : **rsync**, **WinSCP**, etc..).

```
# Debian / Ubuntu
/var/www/wapt/
/var/www/wapt-host/
/var/www/waptwua/
/opt/wapt/conf/
/opt/wapt/waptserver/ssl/

# Centos / RedHat
/var/www/html/wapt/
/var/www/html/wapt-host/
/var/www/html/waptwua/
/opt/wapt/conf/
/opt/wapt/waptserver/ssl/
```

- Sauvegarder la base de données PostgreSQL en utilisant l'utilitaire **pg\_dumpall** (adaptez le nom du fichier à vos besoins).

```
sudo -u postgres pg_dumpall > /tmp/backup_wapt.sql
```

- Redémarrer les services liés à WAPT sur le serveur.

```
systemctl start wapptasks
systemctl start waptserver
systemctl start nginx
```

## 30.2 Windows

- Arrêter les services liés à WAPT sur le serveur.

```
net stop wapptasks
net stop waptserver
net stop waptnginx
```

- Sauvegarder le dossier du dépôt WAPT sur une destination de sauvegarde distante.

```
C:\wapt\conf
C:\wapt\waptserver\repository\wapt
C:\wapt\waptserver\repository\wapt-host
C:\wapt\waptserver\repository\waptwua
C:\wapt\waptserver\nginx\ssl
```

- Sauvegarder la base de données PostgreSQL avec `pg_dump.exe`.

```
"C:\wapt\waptserver\pgsql-9.6\bin\pg_dumpall.exe" -U postgres -f C:\backup_wapt.sql
```

- Redémarrer les services liés à WAPT sur le serveur.

```
net start wapptasks
net start waptserver
net start waptnginx
```

---

## Restauration du serveur WAPT

---

En cas de panne complète, redémarrez une installation standard du serveur WAPT sur votre serveur. Puis suivez cette procédure pour restaurer vos données.

### 31.1 Linux

— Arrêter les services liés à WAPT sur le serveur.

```
systemctl stop nginx
systemctl stop waptserver
systemctl stop wapttasks
```

— Restaurer les répertoires suivants.

```
# Debian / Ubuntu
/var/www/wapt/
/var/www/wapt-host/
/var/www/waptwua/
/opt/wapt/conf/
/opt/wapt/waptserver/ssl/

# Centos / RedHat
/var/www/html/wapt/
/var/www/html/wapt-host/
/var/www/html/waptwua/
/opt/wapt/conf/
/opt/wapt/waptserver/ssl/
```

— Restaurer la base de données (adaptez le nom de votre fichier). La première commande **supprime** la base de données WAPT (si elle existe). Assurez-vous que votre fichier dump est correct avant de le supprimer !

```
sudo -u postgres psql -c "drop database wapt"  
sudo -u postgres psql < /tmp/backup_wapt.sql
```

— Appliquer les droits de propriété aux dossiers restaurés.

```
# Debian / Ubuntu  
chown -R wapt:www-data /var/www/wapt/  
chown -R wapt:www-data /var/www/wapt-host/  
chown -R wapt:www-data /var/www/waptwua/  
chown -R wapt /opt/wapt/conf/  
chown -R wapt /opt/wapt/waptserver/ssl/  
  
# CentOS / RedHat  
chown -R wapt:www-data /var/www/html/wapt/  
chown -R wapt:www-data /var/www/html/wapt-host/  
chown -R wapt:www-data /var/www/html/waptwua/  
chown -R wapt /opt/wapt/conf/  
chown -R wapt /opt/wapt/waptserver/ssl/
```

— Analyser les dépôts de paquets.

```
# Debian / Ubuntu  
wapt-scanpackages /var/www/wapt/  
  
# CentOS / RedHat  
wapt-scanpackages /var/www/html/wapt/
```

— Redémarrer les services liés à WAPT sur le serveur.

```
systemctl start wapttasks  
systemctl start waptserver  
systemctl start nginx
```

## 31.2 Windows

— Arrêter les services liés à WAPT sur le serveur.

```
net start wapttasks  
net start waptserver  
net start waptnginx
```

— Restaurer les répertoires suivants.

```
C:\wapt\waptserver\repository\wapt  
C:\wapt\waptserver\repository\wapt-host  
C:\wapt\waptserver\repository\waptwua  
C:\wapt\waptserver\conf  
C:\wapt\waptserver\nginx\ssl
```

— Appliquer le droit total au dossier C:\wapt\waptserver\repository pour le groupe « Service Réseau ».

— Restaurez la base de données PostgreSQL avec **pg\_restore.exe**.

```
"C:\wapt\waptserver\pgsql-9.6\bin\psql.exe" -f c:\backup_wapt.sql -U postgres
```

- Analyser les dépôts de paquets.

```
wapt-scanpackages "C:\wapt\waptserver\repository\wapt"
```

- Redémarrer les services liés à WAPT sur le serveur.

```
net start wapttasks  
net start waptserver  
net start waptnginx
```





---

## Désinstallation de l'agent WAPT des clients

---

### 32.1 Windows

Si vous devez désinstaller les agents WAPT des clients, le programme de désinstallation est automatiquement créé dans l'emplacement d'installation de WAPT. Par défaut, il s'agit de C:\Program Files (x86)\wapt\unins000.exe.

— La désinstallation silencieuse par défaut d'un agent WAPT peut être réalisée avec la commande suivante.

```
unins000.exe /VERYSILENT
```

— Un argument supplémentaire peut être passé à **unins000.exe** pour tout nettoyer.

```
unins000.exe /VERYSILENT /purge_wapt_dir=1
```

TABLEAU 1 – Liste complète des arguments de ligne de commande pour **unins000.exe**

Paramètres	Description
/VERYSILENT	Lance unins000.exe en silencieux.
/purge_wapt_dir = 1	Purge le répertoire WAPT (supprime tous les dossiers et fichiers).

— Il est possible d'utiliser un paquet pour cela.

```
# -*- coding: utf-8 -*-
from setuphelpers import *

def install():

    print("Creation of the task")
    task = create_onetime_task('removewapt', "unins000.exe", "/VERYSILENT /purge_wapt_dir=1")
    print(task)
```

### 32.1.1 Réactivation des mises à jour de Windows avant la désinstallation

Dans le cas où vous avez utilisé WAPT pour gérer les mises à jour de Windows, vous voudrez peut-être réactiver le comportement par défaut de Windows Updates avant de désinstaller l'agent WAPT.

Pour ce faire, voici un exemple de paquet à pousser avant de désinstaller l'agent WAPT :

```
# -*- coding: utf-8 -*-
from setuphelpers import *

def install():
    print('Disable WAPT WUA')
    inifile_writestring(WAPT.config_filename, 'waptwua', 'enabled', 'false')

    print('DisableWindowsUpdateAccess registry to 0')
    registry_set(HKEY_LOCAL_MACHINE, r'Software\Policies\Microsoft\Windows\WindowsUpdate',
    ↪ 'DisableWindowsUpdateAccess', 0, REG_DWORD)

    print('AUOptions registry to 0')
    registry_set(HKEY_LOCAL_MACHINE, r'SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto_
    ↪ Update', 'AUOptions', 0, REG_DWORD)

    print('Enable wuauserv')
    run_notfatal('sc config wuauserv start= auto')
    run_notfatal('net start wuauserv')

    print('Reload WAPT configuration')
    WAPT.reload_config_if_updated()
```

## 32.2 Linux

— La désinstallation par défaut d'un agent WAPT peut être réalisée avec la commande suivante, selon votre OS Linux :

```
# Debian / Ubuntu
apt remove --purge tis-waptagent

# CentOS / Redhat
yum remove tis-waptagent
```

— Une étape supplémentaire peut être effectuée à l'aide de ces commandes (WIP).

```
# Debian / Ubuntu
rm -f /opt/wapt/
rm /etc/apt/sources.list.d/wapt.list

# CentOS / Redhat
rm -f /opt/wapt/
rm /etc/yum/yum.repos.d/wapt.list
```

## 32.3 MacOS

La désinstallation par défaut d'un agent WAPT peut être réalisée avec la commande suivante :

```
pkgutil --only-files --files it.tranquil.waptservice > file_list  
sudo pkgutil --forget it.tranquil.waptservice
```

Cette section de la documentation couvre l'utilisation quotidienne de WAPT.

Toutes les fonctionnalités de WAPT sont expliquées en détail pour les *Administrateurs*, les *Users* et les *Déployeurs de paquets*.



### 33.1 Windows

Deux méthodes sont disponibles pour déployer **waptagent.exe**.

- La première méthode est manuelle et la procédure doit être appliquée sur chaque machine.
- La seconde est automatisée et repose sur une GPO.

**Note :** L'exécutable d'installation **waptagent.exe** est disponible sur la page d'accueil du site WAPT serveur. Le lien de téléchargement direct est par exemple : <https://srvwapt.mydomain.lan/wapt/waptagent.exe>.

**Avertissement :** Si vous ne signez pas le programme d'installation **waptagent.exe** avec un certificat commercial Code Signing ou un certificat Code Signing émis par l'*autorité de certification* de votre organisation après l'avoir généré, les navigateurs Web afficheront un message d'avertissement lors du téléchargement du programme d'installation. Pour supprimer ce message d'avertissement, vous devez signer le **.exe** avec un certificat *Code Signing* qui peut être vérifié par un faisceau d'autorités de certification stocké dans le magasin de certificats de la machine.

#### 33.1.1 Manuellement

**Attention :** L'installation manuelle nécessite des droits *Administrateur Local* sur l'ordinateur. Installer manuellement l'agent WAPT avec un compte administrateur du domaine **NE MARCHERA PAS**.

**Indication :** Quand déployer l'agent WAPT manuellement ?

La méthode de déploiement manuel est efficace dans ces cas :

- Tester WAPT.
  - Utiliser WAPT dans une organisation comportant un petit nombre d'ordinateurs.
  - Si vous n'avez pas de moyen de déploiement de masse.
- 
- Téléchargez l'agent WAPT depuis votre serveur WAPT puis lancez le programme d'installation.

FIG. 1 – Télécharger l'agent WAPT à déployer sur les ordinateurs

- Choisissez la langue et cliquez sur *OK* pour passer à l'étape suivante.
- Acceptez les conditions de la licence et cliquez sur *Suivant* pour passer à l'étape suivante.
- Choisissez les paramètres supplémentaires et cliquez sur *Suivant* pour passer à l'étape suivante.

TABLEAU 1 – Options disponibles

Paramètres	Description	Valeur par défaut
Installer le service WAPT	Ajouter le service WAPT sur votre ordinateur	Coché
Lancer l'icône de notification lors de l'ouverture de session	Lancer waptagent dans le systray au démarrage	Non coché
Désactiver l'hiberboot, et augmenter le temps pour les GPO (recommandé)	Désactiver le démarrage rapide de Windows pour la stabilité, élargir le délai d'attente pour WAPTExit	Coché
Utiliser un UUID aléatoire pour identifier l'ordinateur au lieu du BIOS	Pour résoudre le <i>bug du BIOS avec l'UUID</i>	Non coché

- Choisissez le dépôt WAPT et le serveur WAPT et cliquez sur *Suivant* pour passer à l'étape suivante.

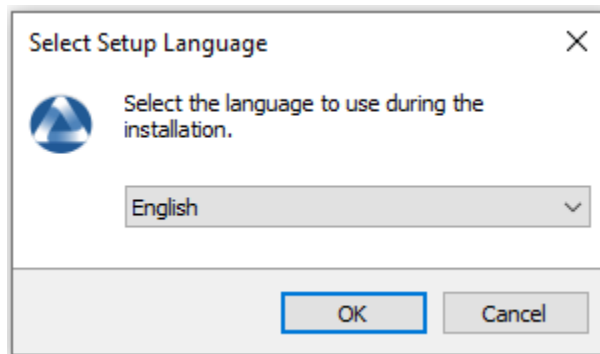


FIG. 2 – Choisissez la langue d'installation

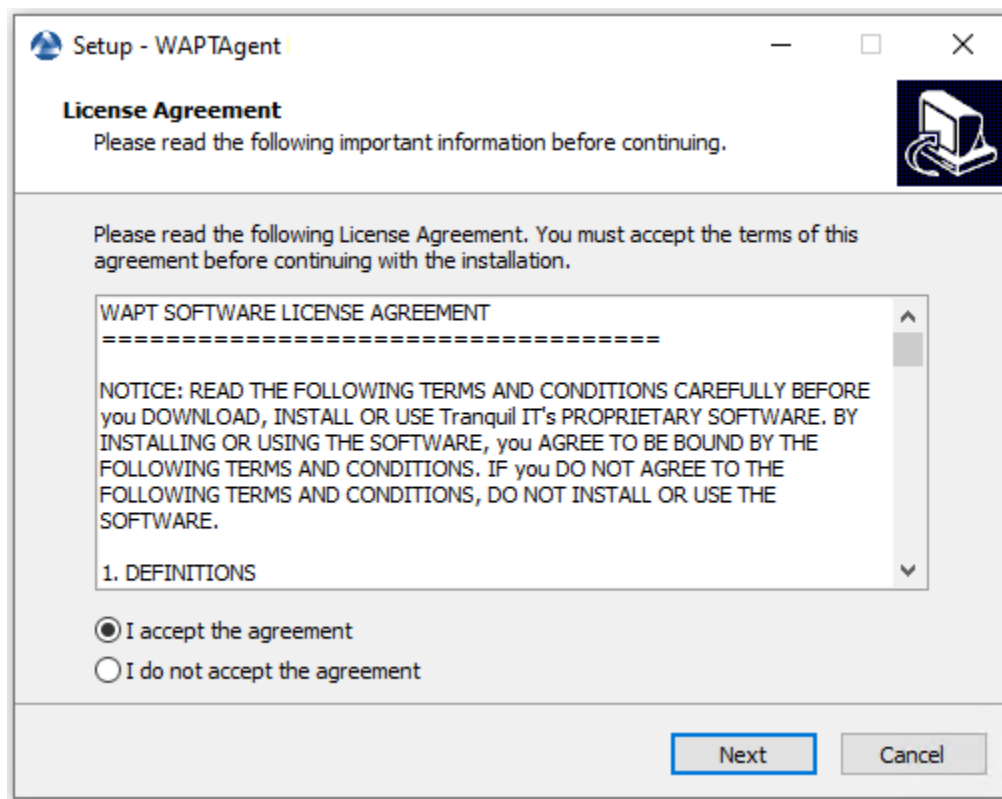


FIG. 3 – Accepter le CLUF

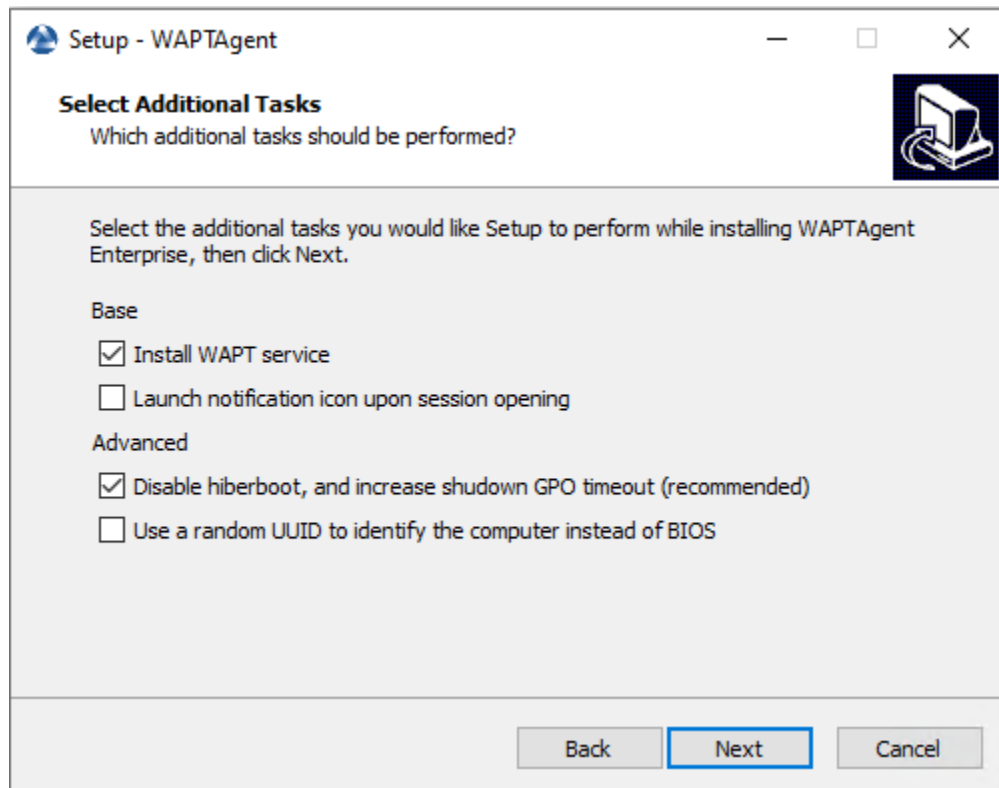


FIG. 4 – Choisissez les options de l'installateur



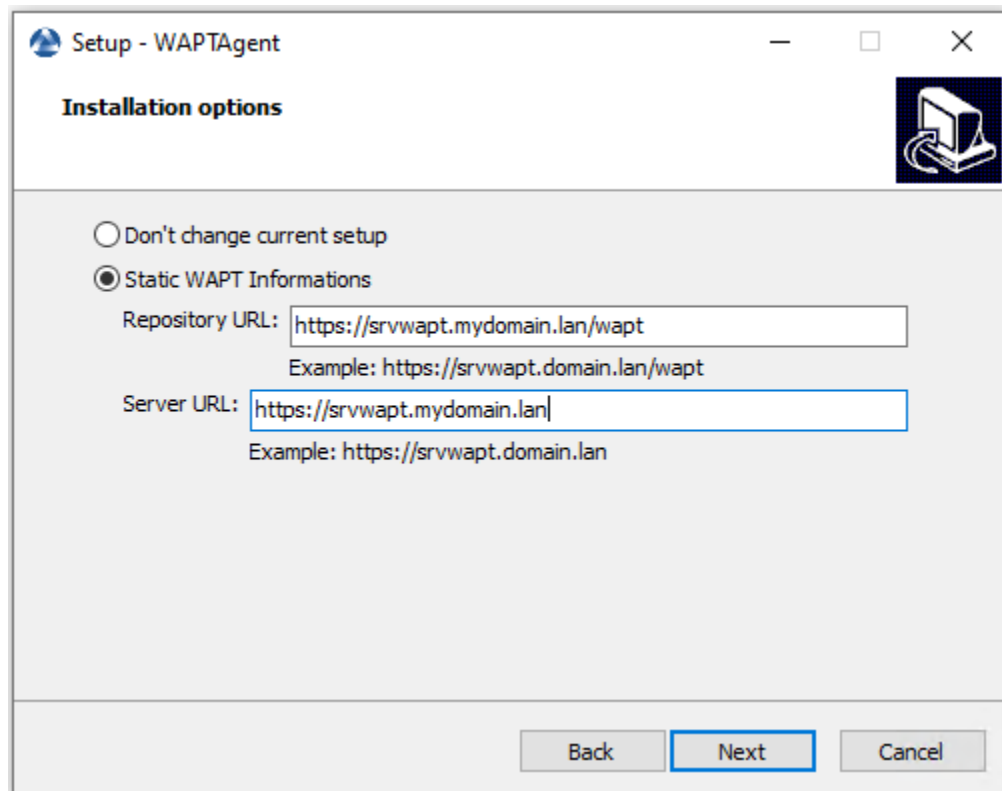


FIG. 5 – Choisir le dépôt et le serveur WAPT

— Installez l'agent WAPT en cliquant sur *Installer* .

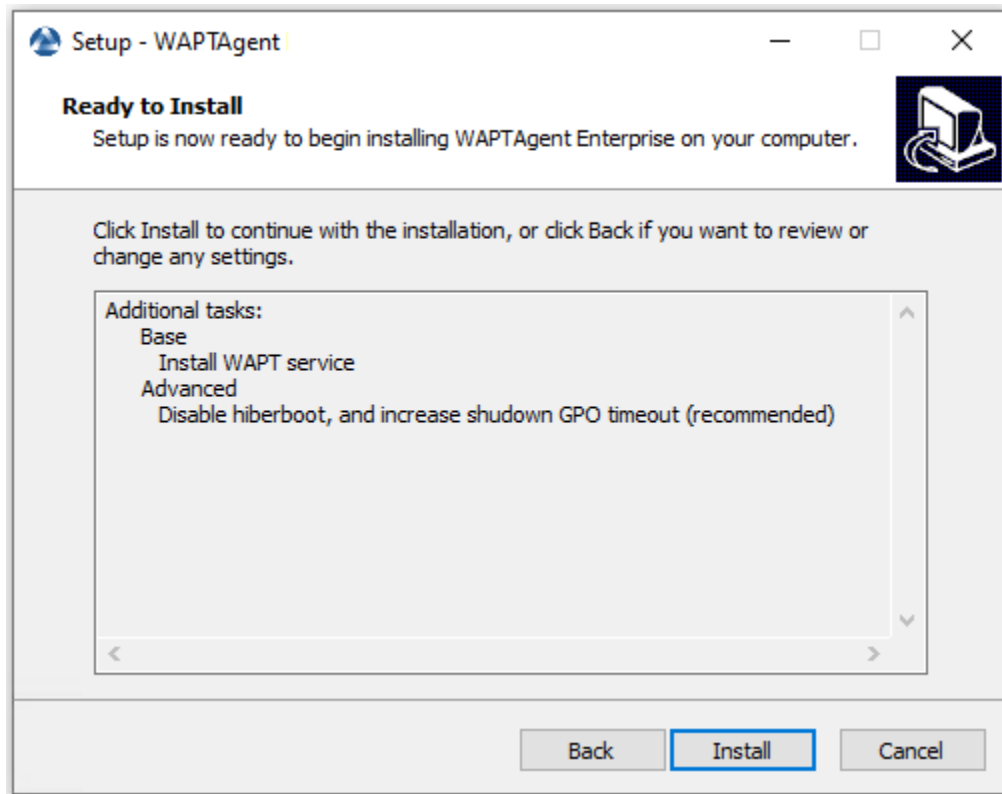


FIG. 6 – Résumé des options d'installation

— Attendez que l'installation de l'agent WAPT se termine, puis cliquez sur *Terminer* pour quitter.

L'installation de l'agent WAPT est terminée. L'enregistrement de la machine auprès du serveur WAPT se fait automatiquement. Pour gérer les clients WAPT de votre organisation, consultez la *documentation sur l'utilisation de la console WAPT*.

### 33.1.2 Automatiquement

---

#### **Important :** Prérequis techniques

Des connaissances avancées en matière d'administration de réseaux et de systèmes sont nécessaires pour mener à bien cette procédure. Un réseau correctement configuré en assurera le succès.

---

#### **Indication :** Quand déployer automatiquement l'agent WAPT ?

La méthode suivante est utile dans ces cas :

- Une grande organisation avec de nombreux ordinateurs.
  - Un Active Directory Samba ou un Active Directory Microsoft pour lequel vous disposez de suffisamment de privilèges d'administration.
  - La sécurité et la traçabilité des actions sont importantes pour vous ou pour votre *Organisation*.
-

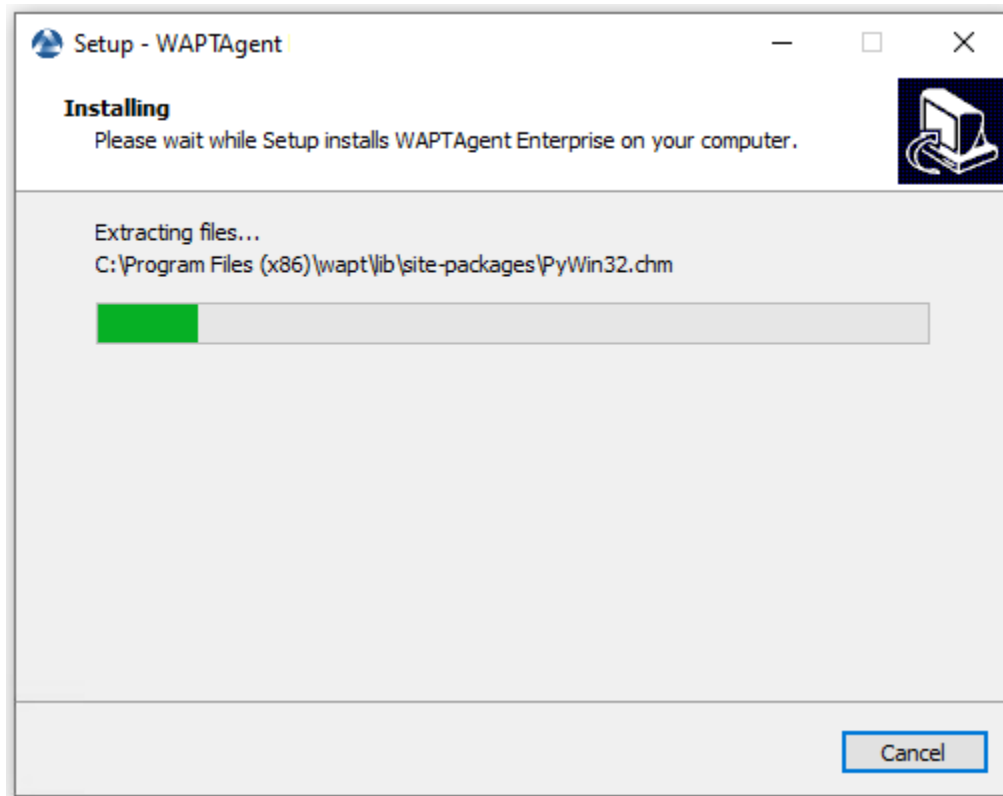


FIG. 7 – Installation en cours

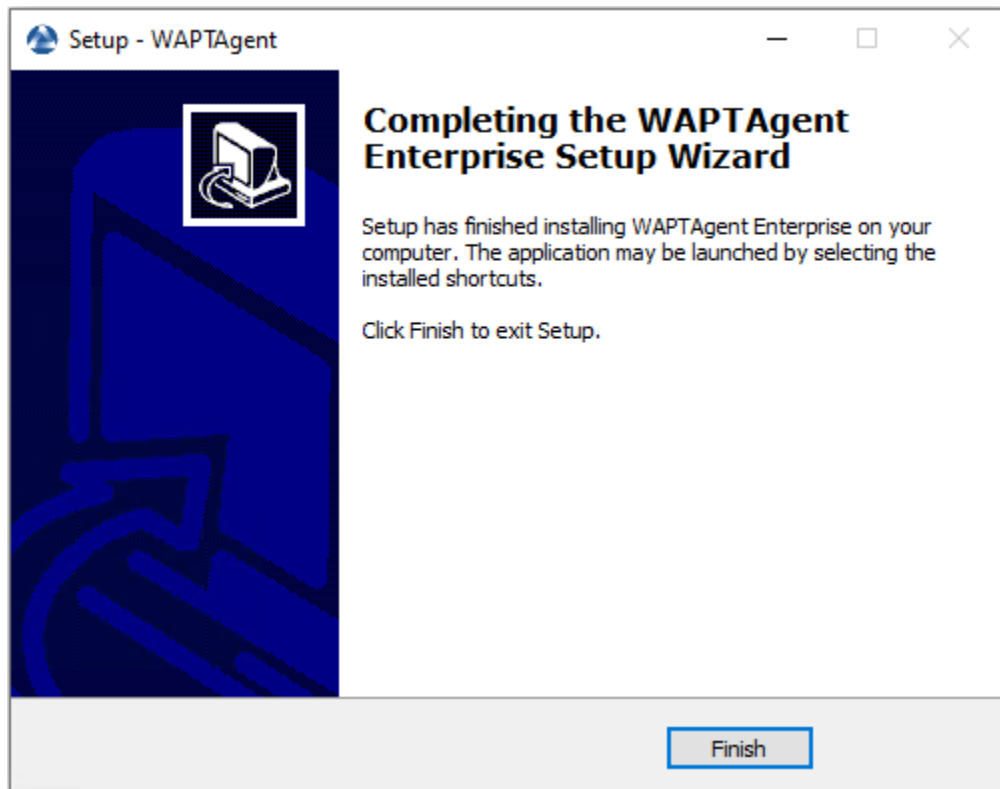


FIG. 8 – Fin de l'installation de l'agent WAPT

## Avec waptdeploy

**waptagent.exe** est un installateur *InnoSetup*, il peut être exécuté avec ces arguments silencieux :

```
waptagent.exe /VERYSILENT
```

— Arguments additionnels disponible pour waptdeploy.

TABLEAU 2 – Description des options disponible pour déployer l’agent WAPT silencieusement

Options	Description
/dnsdomain = mydomain.lan	Domaine dans wapt-get.ini rempli lors de l’installation.
/wapt_server = https://srvwapt.mydomain.lan	URL du dépôt WAPT dans wapt-get.ini rempli pendant l’installation.
/repo_url = https://repo1.mydomain.lan/wapt	URL du dépôt WAPT dans wapt-get.ini rempli pendant l’installation.
/StartPackages = basic-group	Groupe de paquet WAPT installé par défaut.
/verify_cert= 1 ou chemin relatif ssl\server\srvwapt.mydomain.lan.crt.	Valeur de verify_cert entrée lors de l’installation.
/CopyServersTrustedCA = path to a bundle to copy to ssl\server	Paquet de certificats pour les connexions https (à définir par verify_cert).
/CopypackagesTrustedCA = path to a certificate bundle to copy into ssl	Paquet de certificats pour la vérification des signatures de paquets.

**Indication :** Le fichier :file :.\iss` du programme d’installation d’*InnoSetup* est disponible ici : C:\Program Files (x86)\wapt\waptsetup\waptsetup.iss.

Vous pouvez choisir de l’adapter à vos besoins spécifiques. Une fois modifié, il vous suffira de recréer un **waptagent**.

To learn more about the options available with *InnoSetup*, visit this [documentation](#)

**waptdeploy** est un petit binaire qui :

- Vérifie la version de l’agent WAPT.
- Télécharge via https l’installateur **waptagent.exe**.
- Lance le programme d’installation silencieux avec des arguments (options vérifiées définies lors de la compilation de l’agent WAPT).

```
/VERYSILENT /MERGETASKS= ""useWaptServer""
```

— Met à jour le serveur WAPT avec le statut de l’agent WAPT(version WAPT, statut du paquet).

**Avertissement :** **waptdeploy** doit être lancé en tant qu’*Administrateur Local*, c’est pourquoi nous vous recommandons d’utiliser une GPO.

Télécharger waptdeploy.exe depuis votre page d’accueil du serveur WAPT.

**Tranquil IT** DevSecOps

# WAPT Server : ENTERPRISE

Contact Us

WAPT REPOSITORY WAPT SERVER MAILING LIST GESTION DE BUGS (GITHUB) HELP

## WAPT server

WAPT server is managed through a WAPT console installed on a Windows system. When installing the WAPT server on Windows, the console is installed by default and can be found under the start menu.

When installing the server on Linux, the WAPT client should be installed on an administration machine, then run from 'Start/All programs':

To manually add a new host to the WAPT server, download the WAPT agent from the menu to the right. The agent has been properly configured by the server so the default parameters should work. Once the WAPT client has been installed, you can find it in your console.

You can deploy the WAPT agent using a GPO and the WAPT deploy downloader. See [Deployment GPO creation for WAPTDeploy](#)

```
waptdeploy.exe --hash=d988880743bc176680caed24d8fdb64dce9a5dc70c2ca743604b3fc58be2a20 --inversion=2.0.0.9258 --wait=1
```

For further information, be sure to check the documentation at [wapt.fr](#) or on mailing-list.

**Agent WAPT**  
For deploying onto user desktop

- WAPT Server version: 2.0
- WAPT Agent version: 2.0.0.9258
- WAPT Setup version: 2.0.0.9258
- WAPT Deploy version: 2.0.0.9258
- DB status: OK (2.0.3.0)
- Disk space: 50.51 % free

**WAPT Setup**  
For creation of the Wapt agent

**WAPTDeploy**  
For setting up deployment GPO

**Contact**  
[Contact us](#)  
[References](#)  
[News](#)  
[Our team](#)

**Tranquil IT**  
We are a team of passionate people whose life purpose is to be useful to others. We make our products with the aim of resolving your IT problems and optimizing your daily work.

Copyright! Tranquil IT | © 2012-2020

FIG. 9 – Télécharger waptdeploy

## Avec une GPO

- Créer une nouvelle stratégie de groupe sur le serveur Active Directory (Microsoft Active Directory ou Samba-AD).
- Ajouter une nouvelle stratégie : *Configuration de l'ordinateur* → *Stratégies* → *Paramètres Windows* → *Scripts* → *Démarrage* → *Propriétés* → *Ajouter* .

FIG. 10 – Création d'une stratégie de groupe pour déployer l'agent WAPT

- Cliquez sur *Parcourir* pour sélectionner le `waptdeploy.exe`.

FIG. 11 – Rechercher le `waptdeploy.exe` sur votre ordinateur

- Copier `waptdeploy.exe` dans le dossier de destination.

FIG. 12 – Sélection du script `waptdeploy.exe`

- Cliquer sur *Ouvrir* pour importer `waptdeploy.exe`.
- Cliquez sur *Ouvrir* pour confirmer l'importation du binaire **waptdeploy** .

---

**Indication :** Il est nécessaire de donner la somme de contrôle du `waptagent.exe` comme argument au `waptdeploy`. Cela empêchera la machine distante d'exécuter un binaire erroné / corrompu de **waptagent**.

---

```
--hash=checksum WaptAgent --minversion=1.2.3 --wait=15 --waptsetupurl=http://srvwapt.mydomain.lan/  
↪wapt/waptagent.exe
```

Les paramètres et la somme de contrôle de **waptagent.exe** à utiliser pour la GPO `waptdeploy` sont disponibles sur le serveur WAPT en visitant <https://srvwapt.mydomain.lan>.

- Copier les paramètres requis.
- Cliquez sur *OK* pour passer à l'étape suivante.
- Cliquez sur *OK* pour passer à l'étape suivante.
- Appliquer la stratégie GPO résultante aux ordinateurs de l'Organisation OU.

---

**Note :** Nous recommandons d'ajouter `waptdeploy.exe` aux scripts de démarrage sur la GPO.

---



---

**Indication :** Plus d'arguments sont disponible pour `waptdeploy`

---

FIG. 13 – Sélection du script waptdeploy.exe

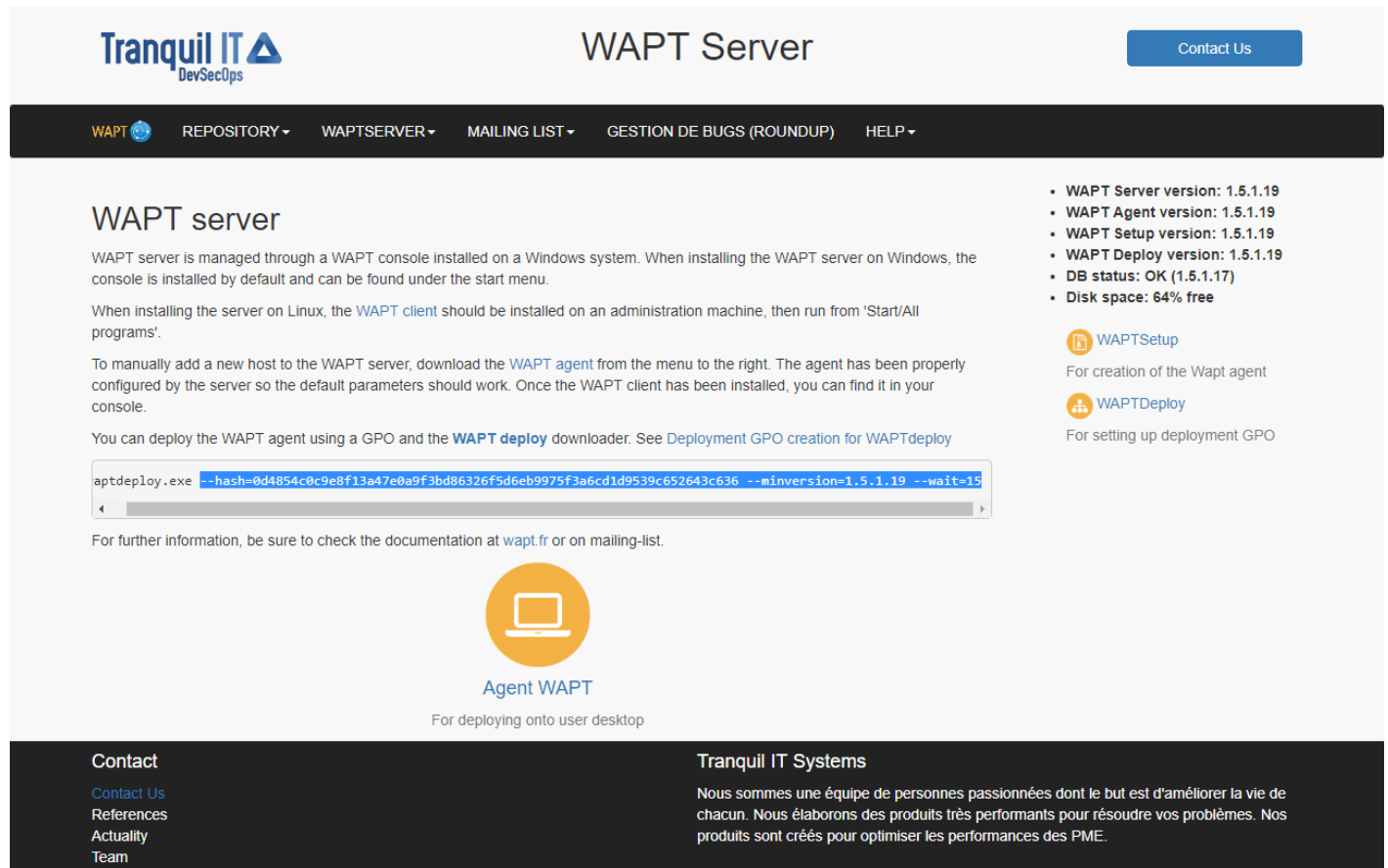


FIG. 14 – Console web du serveur WAPT

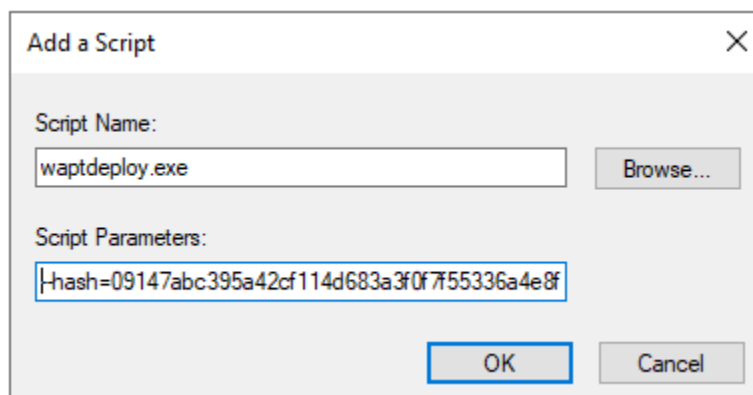


FIG. 15 – ajouter le script waptdeploy à la GPO de démarrage



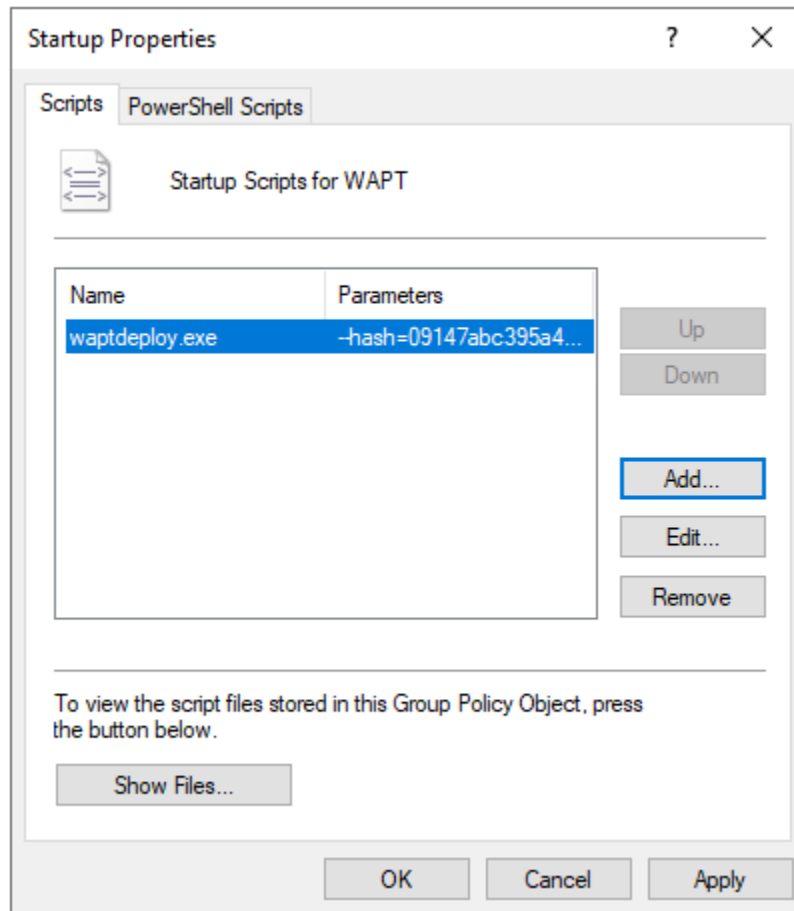


FIG. 16 – La GPO WAPTdeploy sera déployée au prochain démarrage

TABLEAU 3 – Description des options disponibles pour waptdeploy

Options	Description
--force	Installer waptagent.exe même si ce n'est pas nécessaire.
--hash = <sha256hash>	Vérifiez que le hash sha256 du waptagent.exe téléchargé correspond à ce paramètre.
--help	Affiche les options
--minversion = 1.2.3	Installer waptagent.exe si la version installée est inférieure à cela.
--tasks = autorun-Tray,installService,installredis2008,autoUpgradePolicy	Si renseigné, passer les arguments aux options /TASKS de l'installateur de waptagent.exe. Par défaut = installService,installredis2008,autoUpgradePolicy.
--repo_url = https://srvwapt.mydomain.lan/wapt	Emplacement du dépôt pour obtenir waptagent.exe.
--setupargs = <options>	Ajoutez ceci à la ligne de commande de waptagent.exe.
--wait = <minutes>	Attendez que les tâches en cours d'exécution et en attente se terminent si waptservice est en cours d'exécution avant l'installation.
--waptsetupurl = https://srvwapt.mydomain.lan/wapt/waptagent.exe	Emplacement explicite pour télécharger l'exécutable d'installation. Peut être un chemin local (par défaut= <repo_url>/waptagent.exe.

## Avec une tâche planifiée

Vous pouvez également choisir de lancer **waptdeploy** en utilisant une tâche planifiée qui a été définie par GPO.

**Indication :** Cette méthode est particulièrement efficace pour déployer WAPT sur des postes de travail lorsque le réseau n'est pas disponible au démarrage ou à l'arrêt.

La méthode consiste à utiliser une GPO pour copier localement waptdeploy.exe et waptagent.exe et créer une tâche planifiée pour l'installation.

- Copiez waptdeploy.exe et waptagent.exe dans le partage netlogon de votre serveur Active Directory (\mydomain.lan\netlogon\waptagent.exe).
- Créer une nouvelle stratégie de groupe sur le serveur Active Directory (Microsoft Active Directory ou Samba-AD).
- Ajouter une nouvelle stratégie : *Configuration ordinateur* → *Préférences* → *Paramètres Windows* → *Fichiers*.
- Créer un nouveau fichier pour copier waptdeploy.
- Définir les paramètres.

TABLEAU 4 – Description des options pour la copie

Options	Valeur
Action	Remplacer
Fichier(s) source	\mydomain.lan\netlogon\waptdeploy.exe
Fichier de destination	C:\Temp\waptdeploy.exe
Suppression des erreurs sur les actions de fichiers individuels	non coché
Lecture seule	non coché
Masqué	non coché
Archiver	coché

- Créer un nouveau fichier pour copier waptagent.
- Définir les paramètres.

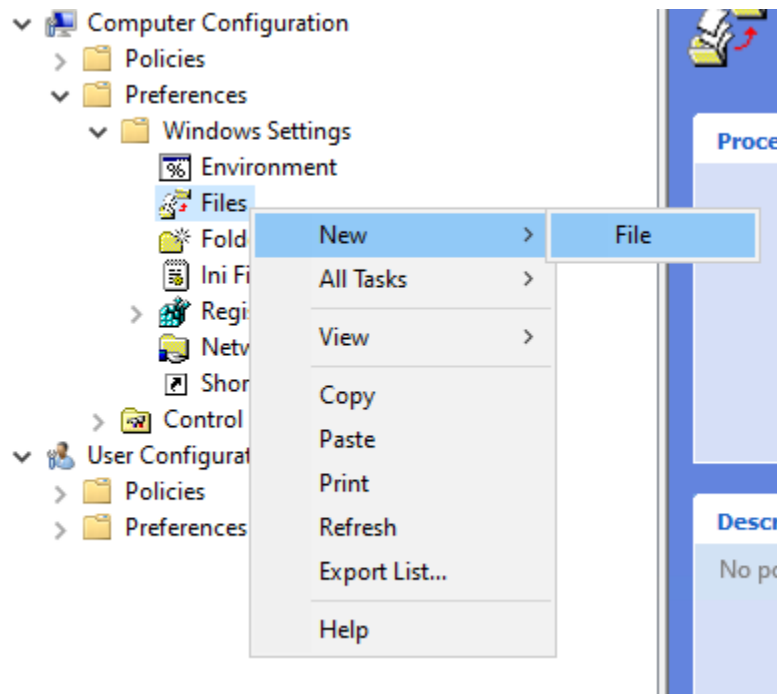


FIG. 17 – Nouveau fichier dans la GPO

TABLEAU 5 – Description des options pour la copie

Options	Valeur
Action	Remplacer
Fichier(s) source	\\mydomain.lan\netlogon\waptagent.exe
Fichier de destination	C:\Temp\waptagent.exe
Suppression des erreurs sur les actions de fichiers individuels	non coché
Lecture seule	non coché
Masqué	non coché
Archiver	coché

- Puis aller dans le menu des tâches planifiées : *Configuration ordinateur* → *Préférences* → *Panneau de configuration* → *Tâches planifiées*.
- Créer une nouvelle tâche planifiée : *Clic droit* → *Nouveau* → *Tâche planifiée* (Au moins Windows 7).
- Définir *Action* sur *Remplacer*.
- Pour *Lorsque la tâche est en cours*, utiliser le compte suivant collez *S-1-5-18* (compte système). Vous pouvez visiter pour plus d'information.
- Cocher *Exécuter si l'utilisateur est connecté ou non*.
- Cocher *Exécuter avec les privilèges les plus élevés*, puis aller dans l'onglet *Déclencheur*.
- Créer un nouveau déclencheur.
- Cocher *Tous les jours*, select la date du jour.
- Cocher *Répéter la tâche tous les* et sélectionnez *1 heure* et pour une durée de sélectionnez *1 jour*.
- Cocher *Arrêter la tâche si elle dure plus de* et sélectionnez *2 heures*.
- Vérifier que *Enabled* est coché, puis aller dans l'onglet *Actions*.
- Créer une nouvelle action *Démarrer un programme* pour *waptdeploy.exe*.

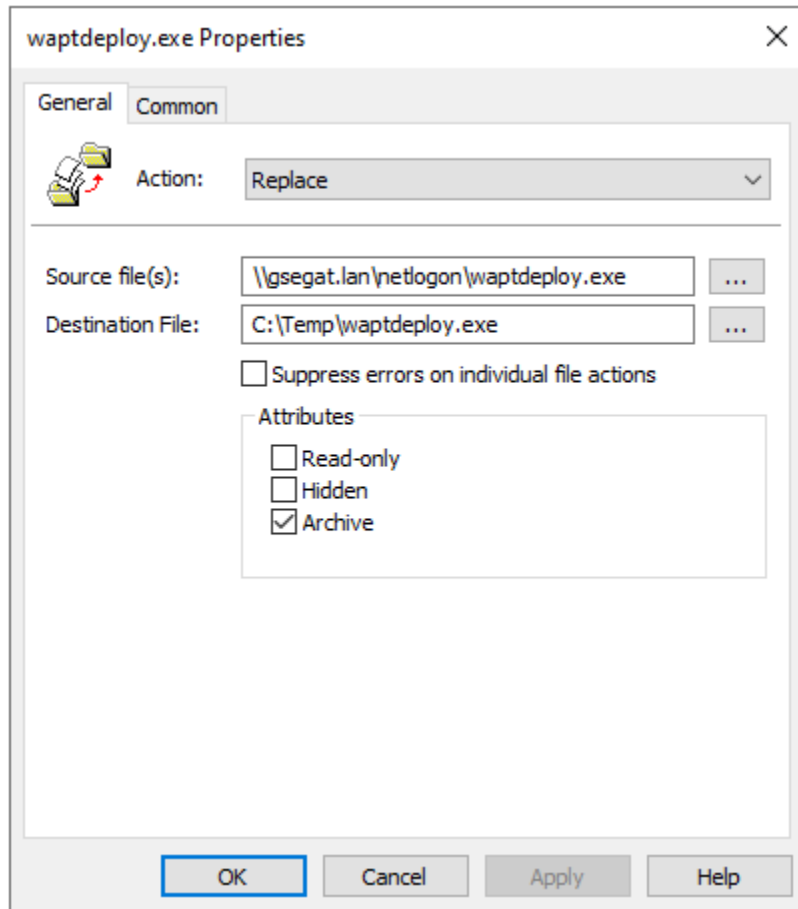


FIG. 18 – Progression de l'installation de l'agent WAPT

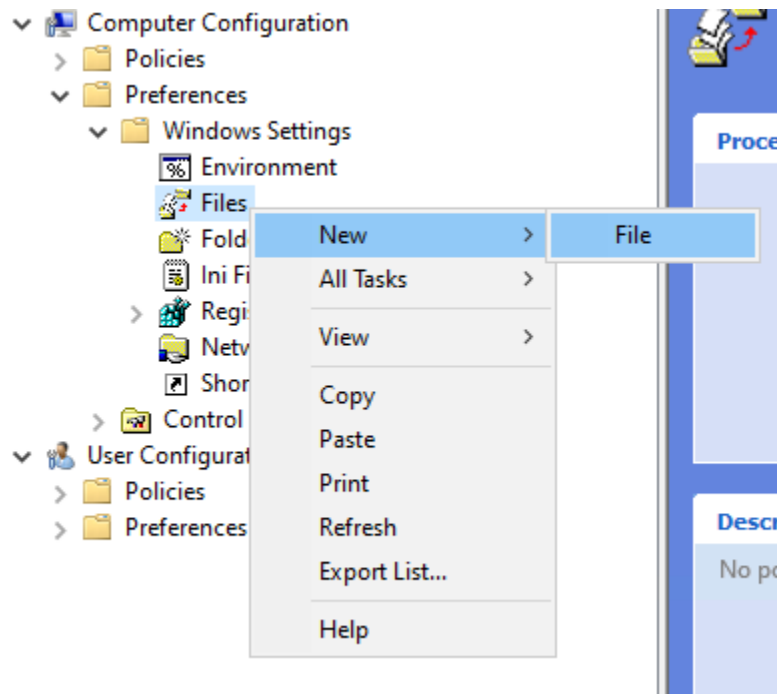


FIG. 19 – Nouveau fichier dans la GPO

TABLEAU 6 – Description des options pour la copie

Options	Valeur
Action	Démarrer un programme
Programme / script	C:\Temp\waptagent.exe
Ajouter arguments (facultatif)	Voir le point suivant
Commence dans (facultatif)	vide

**Indication :** Il est nécessaire de renseigner la somme de contrôle du `waptagent.exe` comme argument au `waptdeploy`. Cela empêchera la machine distante d'exécuter un binaire erroné / corrompu de `waptagent`.

```
--hash=checksum WaptAgent --minversion=1.2.3 --wait=15 --waptsetupurl=http://srvwapt.mydomain.lan/  
↪wapt/waptagent.exe
```

Les paramètres et la somme de contrôle `waptagent.exe` à utiliser pour la GPO `waptdeploy` sont disponibles sur le serveur WAPT en visitant <https://srvwapt.mydomain.lan>.

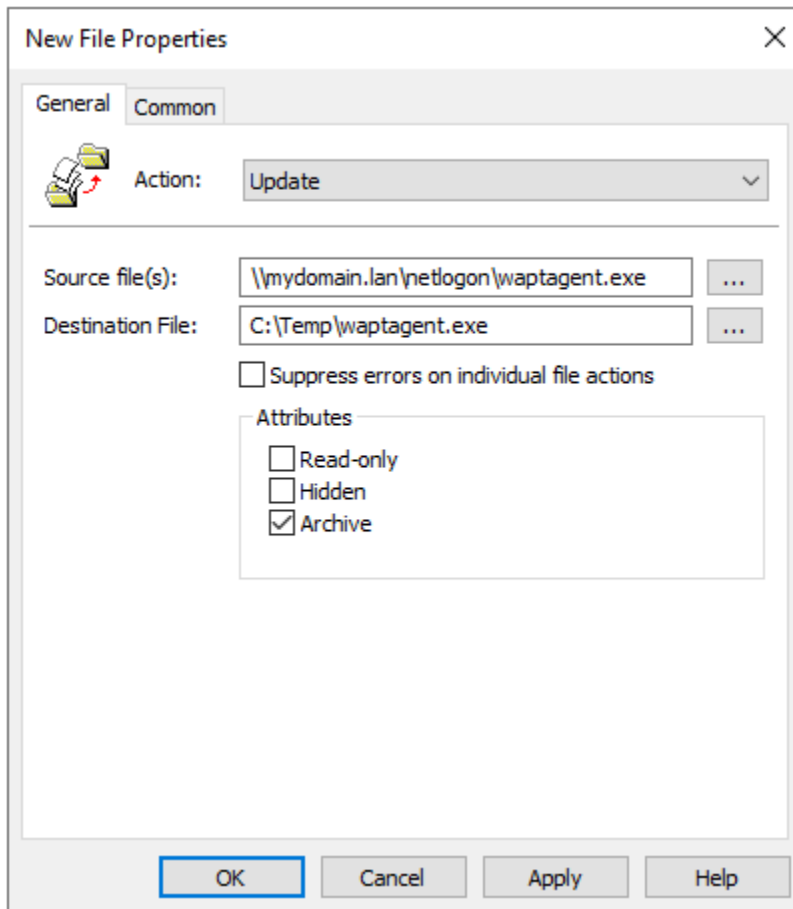


FIG. 20 – Progression de l'installation de l'agent WAPT

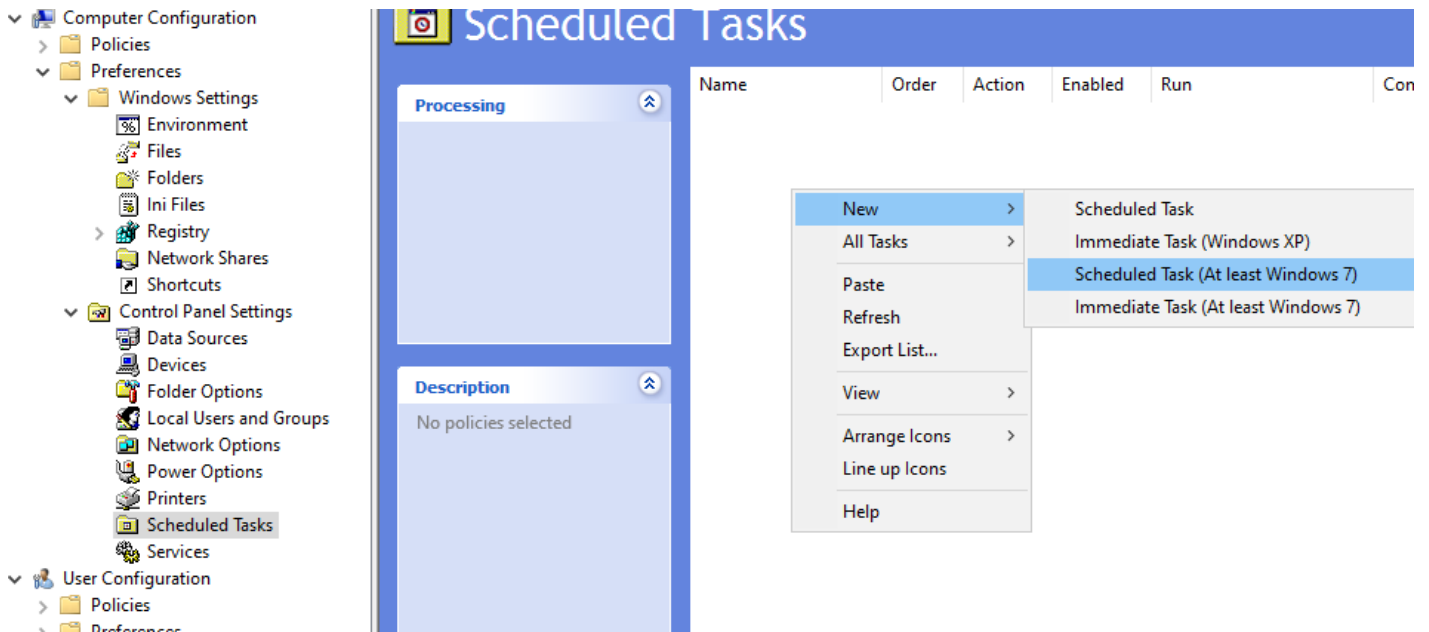


FIG. 21 – Tâche Créer dans la fenêtre des propriétés de *deploywapt*

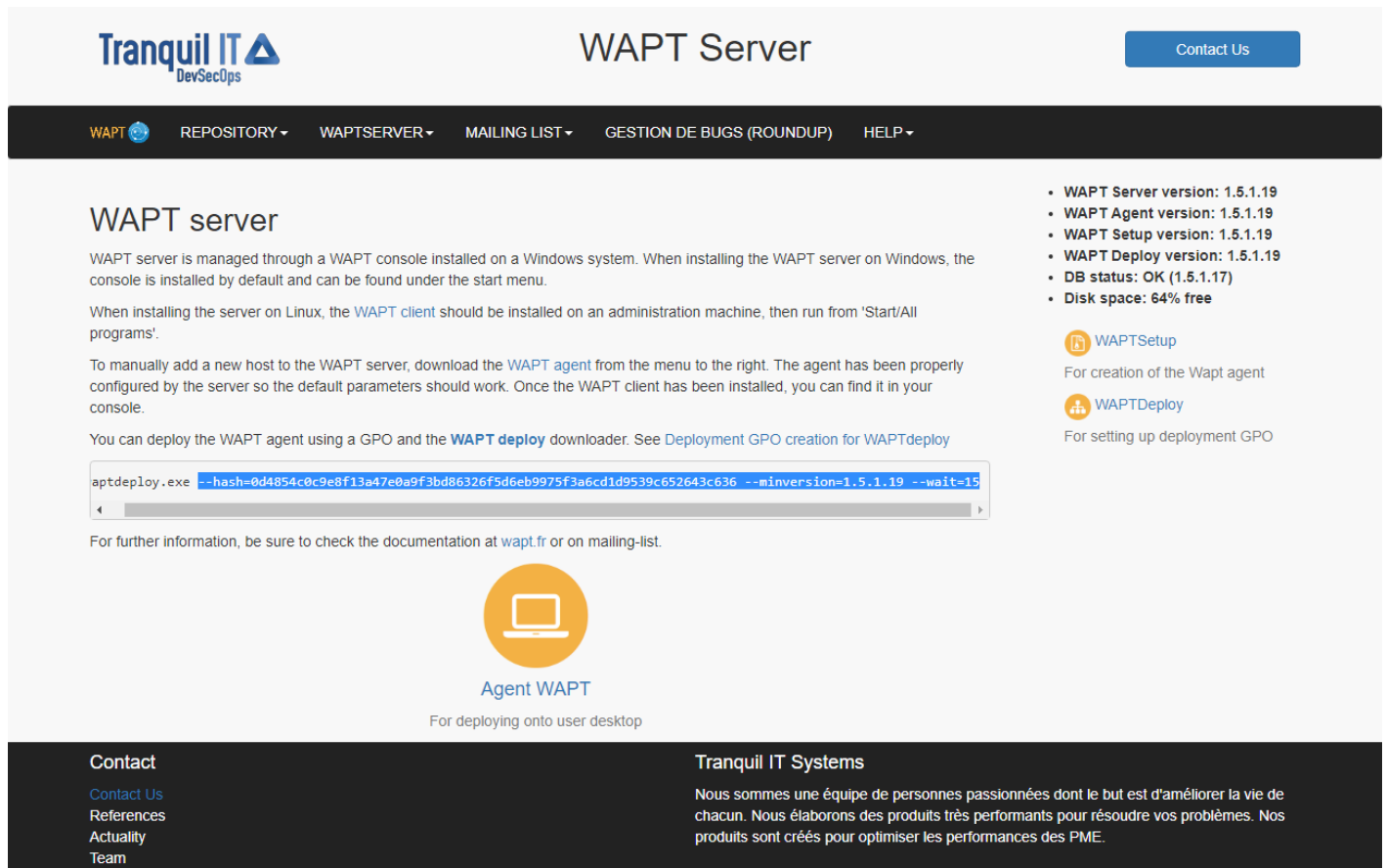


FIG. 24 – Console web du serveur WAPT

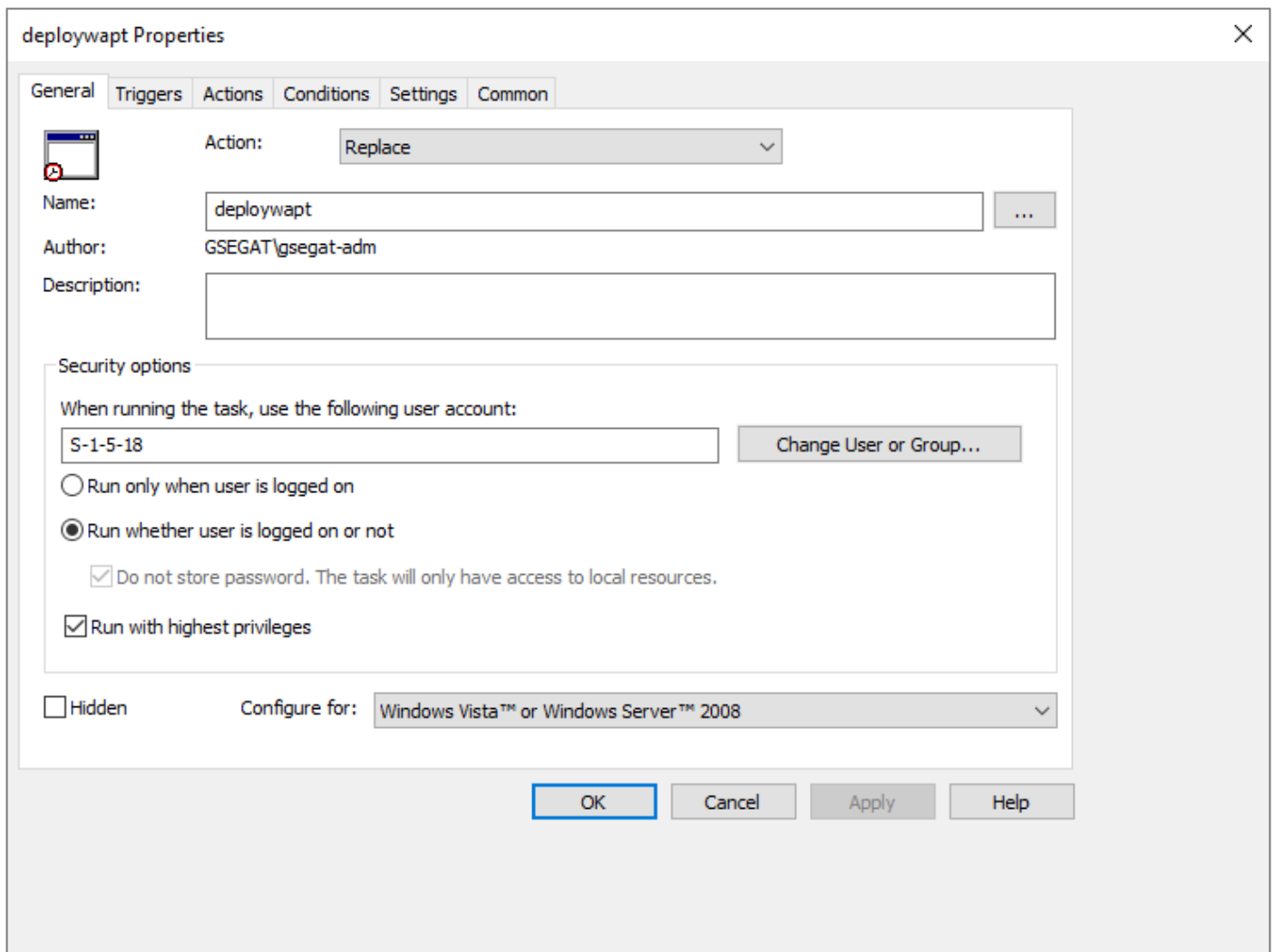


FIG. 22 – Onglet Général de la fenêtre de propriété de *deploywapt*



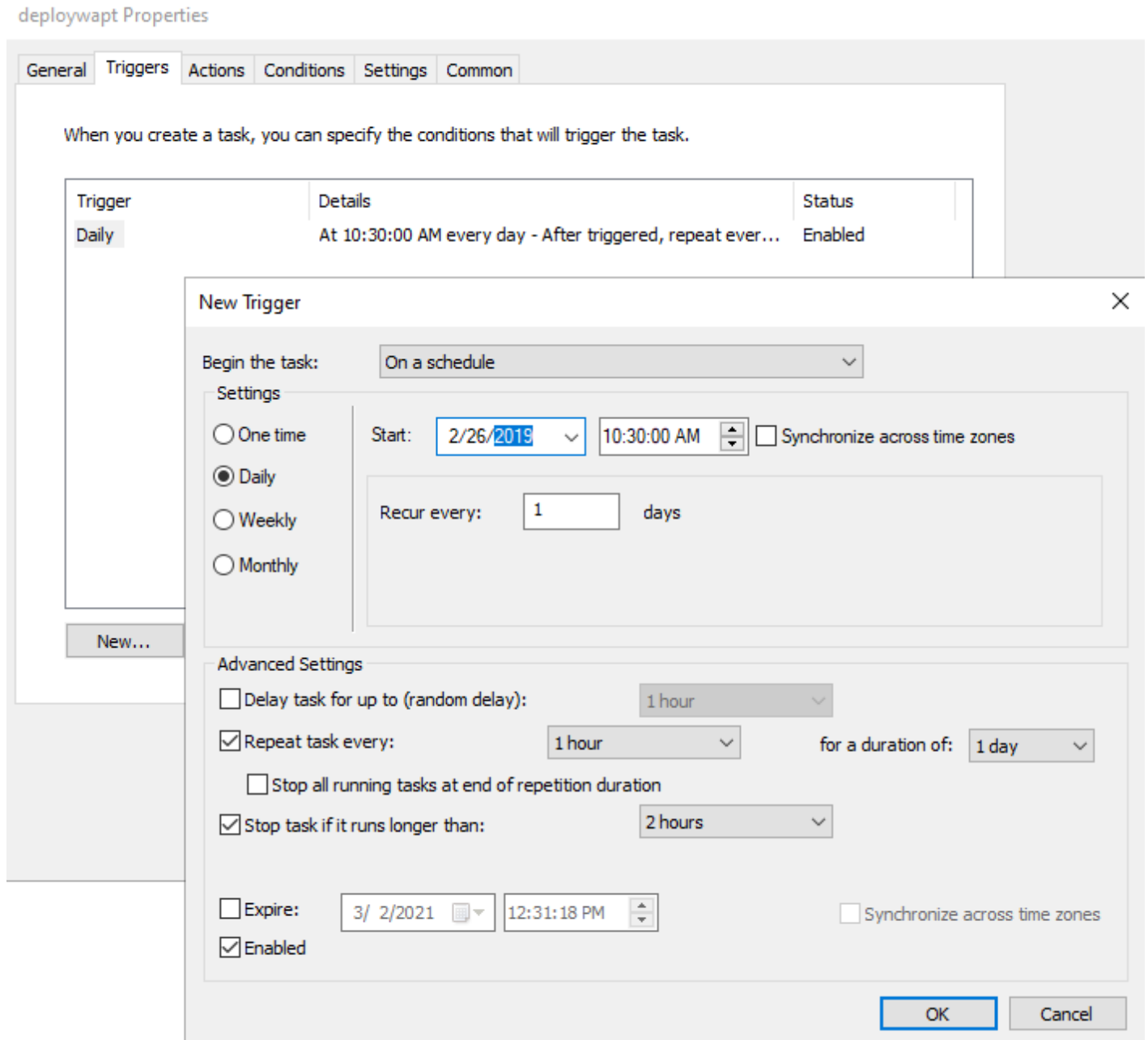
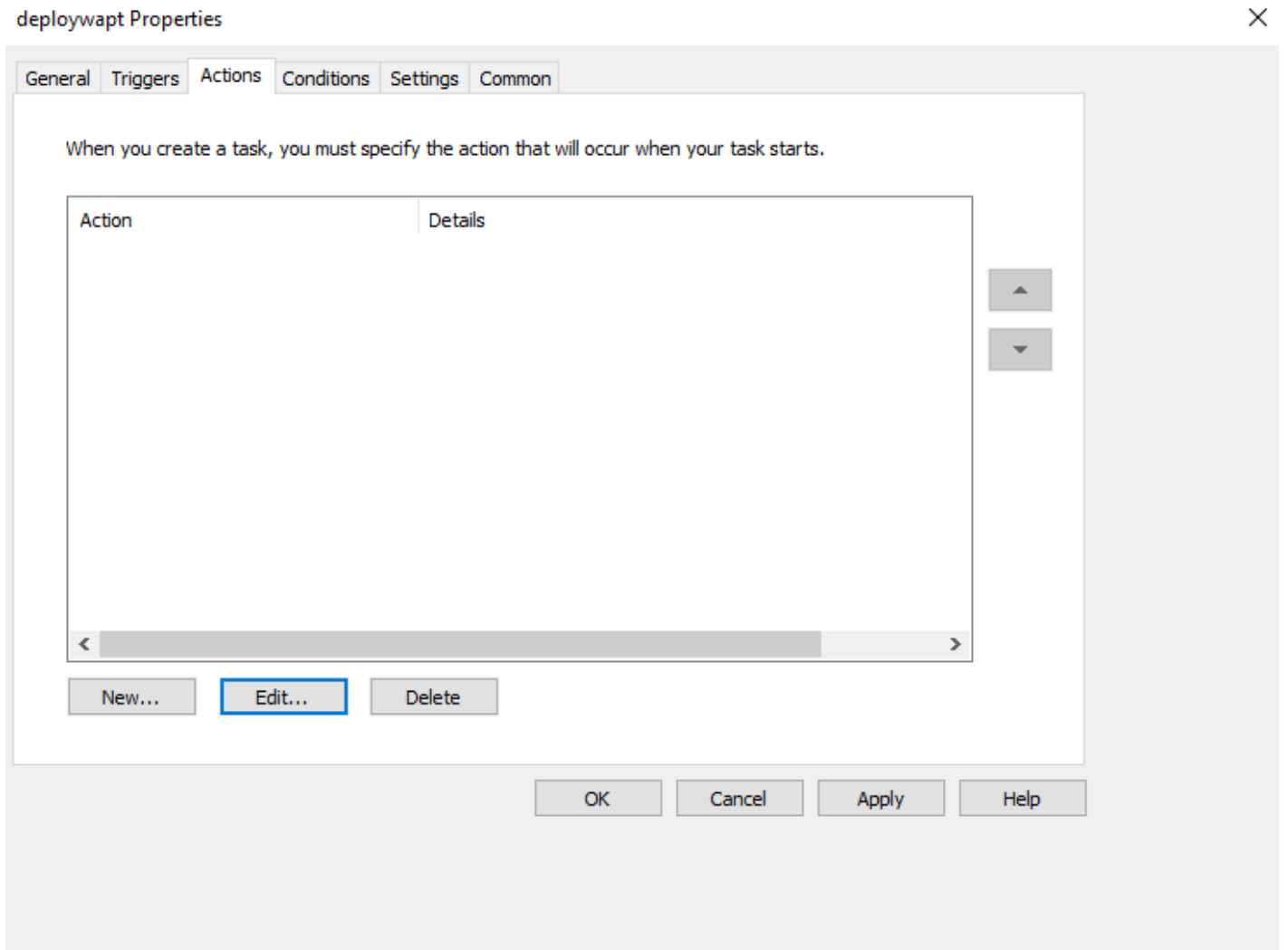


FIG. 23 – Onglet Déclencheur de la fenêtre de propriété *deploywapt*



— Copiez les paramètres requis et changez waptsetupurl à C:\Temp\waptagent.exe.

```
--hash=checksum WaptAgent --minversion=1.2.3 --wait=15 --waptsetupurl=C:\Temp\waptagent.exe
```

**Indication :** Plus d'arguments sont disponible pour waptdeploy

TABLEAU 7 – Description des options disponibles pour waptdeploy

Options	Description
--force	Installer waptagent.exe même si ce n'est pas nécessaire
--hash = <sha256hash>	Vérifiez que le hash sha256 du waptagent.exe téléchargé correspond à ce paramètre.
--help	Affiche les options.
--minversion = 1.2.3	Installer waptagent.exe si la version installée est inférieure à cela.
--tasks = autorun-Tray,installService,installredis2008,autoUpgradePolicy	Si renseignés, donner les arguments aux options /TASKS de l'installateur de waptagent.exe. Par défaut = installService,installredis2008,autoUpgradePolicy
--repo_url = https://srvwapt.mydomain.lan/wapt	Emplacement du dépôt où obtenir waptagent.exe.
--setupargs = <options>	Ajoutez ceci à la ligne de commande de waptagent.exe.
--wait = <minutes>	Attendez que les tâches en cours d'exécution et en attente se terminent si waptservice est en cours d'exécution.
--waptsetupurl = https://srvwapt.mydomain.lan/wapt/waptagent.exe	L'emplacement explicite pour télécharger l'exécutable de configuration. Peut être un chemin local (par défaut= <repo_url>/waptagent.exe).

— Aller à l'onglet *Paramètres*.

— Dans l'onglet *Paramètres*, vérifier seulement *Exécuter la tâche dès que possible après qu'un démarrage planifié soit manquant*.

**Indication :** Pour vérifier que votre GPO fonctionne, vous pouvez exécuter la commande **gpupdate /force** et vérifier que la tâche de planification est présente sur votre ordinateur en lançant **Planificateur de tâches** en mode administrateur local.

## 33.2 Linux

Un agent Linux est disponible pour *Debian*, *Ubuntu* et *RedHat / Centos*.

**Note :** La procédure suivante installe un agent WAPT en utilisant les dépôts de Tranquil IT ;

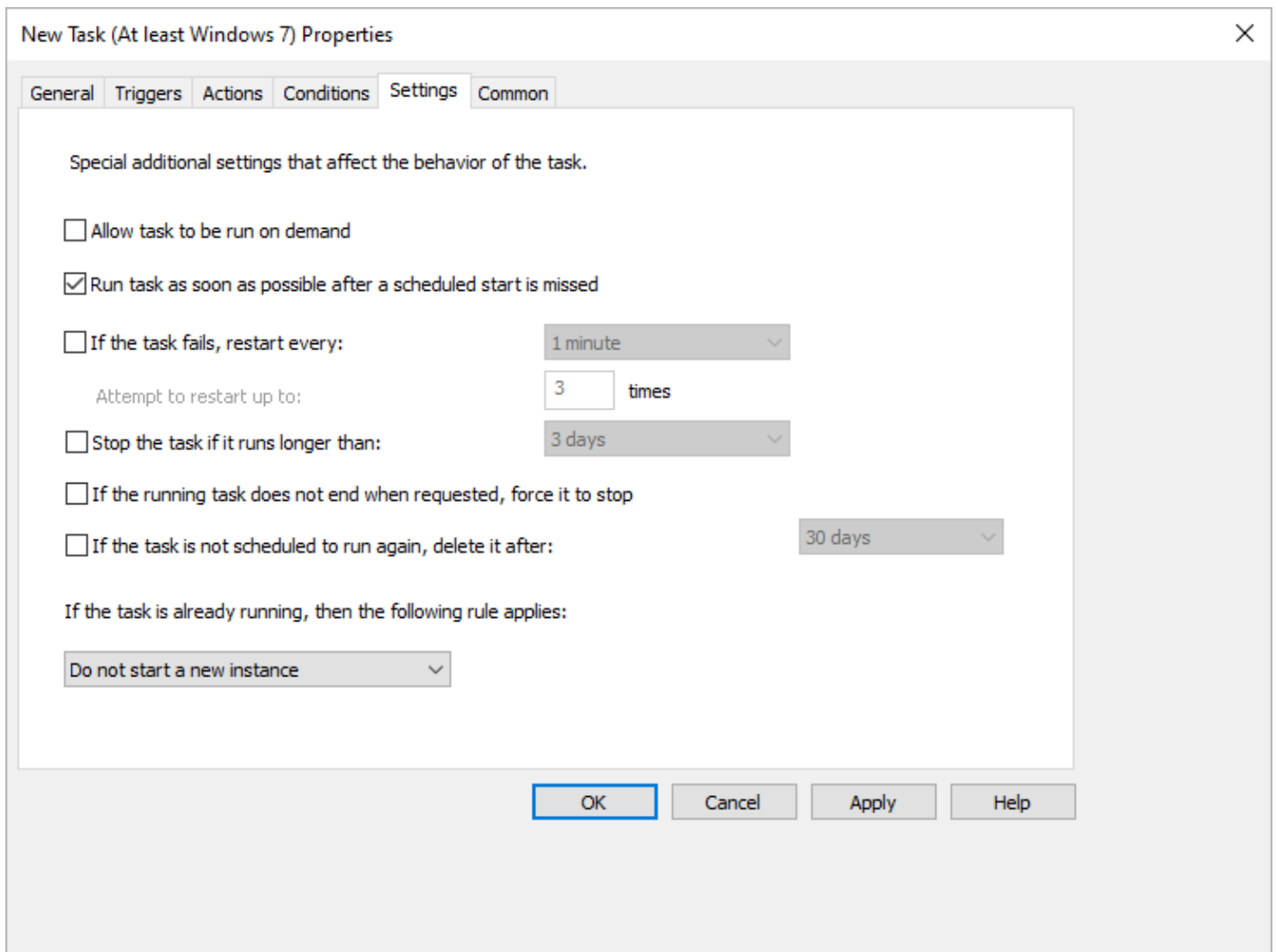


FIG. 25 – Onglet Paramètres de la fenêtre de propriété *deploywapt*

### 33.2.1 Debian

#### Discovery

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>

#### Enterprise

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Enterprise**. Pour WAPT **Discovery** Edition, veuillez vous référer au bloc précédent.

— Mise à jour de la distribution :

```
apt update && apt upgrade -y
```

— Installez apt-transport-https pour l'utilisation de https :

```
apt install apt-transport-https lsb-release gnupg -y
```

— Récupération de la clé .gpg et ajout dans le dépôt Tranquil'iT :

```
wget -O - https://wapt.tranquil.it/debian/tiswapt-pub.gpg | apt-key add -
echo "deb https://srvwapt-pro.tranquil.it/entreprise/debian/wapt-2.1/ $(lsb_release -c -s) main" > /
→etc/apt/sources.list.d/wapt.list
```

— Créez wapt.conf dans /etc/apt/auth.conf.d pour stocker vos informations de connexion

**Indication :** Remplacez **user** et **password** pour accéder au référentiel WAPT Enterprise, par ceux fournis par notre service commercial.

```
cat > /etc/apt/auth.conf.d/wapt.conf <<EOF
machine srvwapt-pro.tranquil.it
login user
password password
EOF
```

— Appliquer les droits suivants

```
chmod 600 /etc/apt/auth.conf.d/wapt.conf
```

— installer l'agent WAPT en utilisant apt-get :

```
export DEBIAN_FRONTEND=noninteractive
apt update
apt install tis-waptagent -y
unset DEBIAN_FRONTEND
```

### Création du fichier de configuration de l'agent

---

**Indication :** Utilisez l'adresse de votre serveur dans `repo_url` et `wapt_server`.

---

```
cat > /opt/wapt/wapt-get.ini <<EOF
[global]
repo_url=https://srvwapt.mydomain.lan/wapt
wapt_server=https://srvwapt.mydomain.lan
use_hostpackages=1
use_kerberos=0
verify_cert=0
EOF
```

### Copie du certificat de signature du paquet

Vous devez copier manuellement, ou par script, le certificat public de votre autorité de certification de signature de paquet.

Le certificat doit se trouver sur votre machine Windows dans `C:\Program Files (x86)\wapt\ssl\`.

Copiez votre ou vos certificats dans `/opt/wapt/ssl` en utilisant **WinSCP** ou **rsync** si vous déployez sur Linux ou MacOS.

### Copie du certificat SSL/TLS

Si vous avez déjà configuré votre serveur WAPT pour utiliser les *certificats SSL/TLS avec Nginx*, vous devez copier le certificat dans votre agent WAPT Linux.

Le certificat doit se trouver sur votre ordinateur Windows, dans `C:\Program Files (x86)\wapt\ssl\server\`.

- Copiez votre ou vos certificats dans `/opt/wapt/ssl/server/` en utilisant **WinSCP** ou **rsync** si vous déployez sur Linux ou MacOS.
- Ensuite, modifiez dans votre fichier de configuration `/opt/wapt/wapt-get.ini` le chemin vers votre certificat.

---

**Indication :** Changez `YOURCERT.crt` par le nom de votre certificat.

---

```
verify_cert=/opt/wapt/ssl/server/YOURCERT.crt
```

## Enregistrer votre hôte Linux sur le serveur WAPT

- Redémarrez le service WAPT :

```
systemctl restart waptservice.service
```

- Enfin, exécutez la commande suivante pour enregistrer votre hôte Linux avec le serveur WAPT.

```
wapt-get register  
wapt-get update
```

- Redémarrez le service WAPT après l'enregistrement :

```
systemctl restart waptservice.service
```

**Félicitations**, votre agent Linux est maintenant installé et configuré et il apparaîtra maintenant dans votre console WAPT avec une icône en forme de pingouin.

### Fonctions non prises en charge

- Installation des mises à jour à l'arrêt.
- La console WAPT n'est pas actuellement disponible sur Linux.
- Une fonction spécifique à Windows.

### Particularités de la fonctionnalité du domaine

- Les tests ont été effectués avec sssd avec un domaine Active Directory et une authentification kerberos.
- Pour intégrer une machine dans le domaine Active Directory, vous pouvez choisir de suivre [cette documentation](#).
- Pour forcer la mise à jour des unités d'organisation sur l'hôte, vous pouvez appliquer un **gpupdate** à partir de la console WAPT.
- Pour que les groupes Active Directory fonctionnent correctement, vous devez vérifier que la commande **id hostname\$** renvoie la liste des groupes dont l'hôte est membre.

**Attention :** Nous avons remarqué que la requête LDAP de kerberos ne fonctionne pas si le reverse DNS record n'est pas configuré correctement pour vos contrôleurs de domaine. Ces enregistrements doivent donc être créés s'ils n'existent pas.

## 33.2.2 Ubuntu

### Discovery

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>

---

### Enterprise

---

**Important : Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT Enterprise.** Pour WAPT Discovery Edition, veuillez vous référer au bloc précédent.

---

— Mise à jour de la distribution :

```
apt update && apt upgrade -y
```

— Installez `apt-transport-https` pour l'utilisation de https :

```
apt install apt-transport-https lsb-release gnupg -y
```

— Récupération de la clé `.gpg` et ajout dans le dépôt Tranquil'iT :

```
wget -O - https://wapt.tranquil.it/debian/tiswapt-pub.gpg | apt-key add -  
echo "deb https://srvwapt-pro.tranquil.it/entreprise/ubuntu/wapt-2.1/ $(lsb_release -c -s) main" > /  
↪etc/apt/sources.list.d/wapt.list
```

— Créez `wapt.conf` dans `/etc/apt/auth.conf.d` pour stocker vos informations de connexion

---

**Indication :** Remplacez `user` et `password` pour accéder au référentiel WAPT Enterprise, par ceux fournis par notre service commercial.

---

```
cat > /etc/apt/auth.conf.d/wapt.conf <<EOF  
machine srvwapt-pro.tranquil.it  
login user  
password password  
EOF
```

— Appliquer les droits suivants

```
chmod 600 /etc/apt/auth.conf.d/wapt.conf
```

— installer l'agent WAPT en utilisant `apt-get` :

```
export DEBIAN_FRONTEND=noninteractive  
apt update  
apt install tis-waptagent  
unset DEBIAN_FRONTEND
```



## Création du fichier de configuration de l'agent

**Indication :** Utilisez l'adresse de votre serveur dans `repo_url` et `wapt_server`.

```
cat > /opt/wapt/wapt-get.ini <<EOF
[global]
repo_url=https://srvwapt.mydomain.lan/wapt
wapt_server=https://srvwapt.mydomain.lan
use_hostpackages=1
use_kerberos=0
verify_cert=0
EOF
```

## Copie du certificat de signature du paquet

Vous devez copier manuellement, ou par script, le certificat public de votre autorité de certification de signature de paquet.

Le certificat doit se trouver sur votre machine Windows dans `C:\Program Files (x86)\wapt\ssl\`.

Copiez votre ou vos certificats dans `/opt/wapt/ssl` en utilisant **WinSCP** ou **rsync** si vous déployez sur Linux ou MacOS.

## Copie du certificat SSL/TLS

Si vous avez déjà configuré votre serveur WAPT pour utiliser les *certificats SSL/TLS avec Nginx*, vous devez copier le certificat dans votre agent WAPT Linux.

Le certificat doit se trouver sur votre ordinateur Windows, dans `C:\Program Files (x86)\wapt\ssl\server\`.

- Copiez votre ou vos certificats dans `/opt/wapt/ssl/server/` en utilisant **WinSCP** ou **rsync** si vous déployez sur Linux ou MacOS.
- Ensuite, modifiez dans votre fichier de configuration `/opt/wapt/wapt-get.ini` le chemin vers votre certificat.

**Indication :** Changez `YOURCERT.crt` par le nom de votre certificat.

```
verify_cert=/opt/wapt/ssl/server/YOURCERT.crt
```

## Enregistrer votre hôte Linux sur le serveur WAPT

- Redémarrez le service WAPT :

```
systemctl restart waptservice.service
```

- Enfin, exécutez la commande suivante pour enregistrer votre hôte Linux avec le serveur WAPT.

```
wapt-get register
wapt-get update
```

- Redémarrez le service WAPT après l'enregistrement :

```
systemctl restart waptservice.service
```

**Félicitations**, votre agent Linux est maintenant installé et configuré et il apparaîtra maintenant dans votre console WAPT avec une icône en forme de pingouin.

### Fonctions non prises en charge

- Installation des mises à jour à l'arrêt.
- La console WAPT n'est pas actuellement disponible sur Linux.
- Une fonction spécifique à Windows.

### Particularités de la fonctionnalité du domaine

- Les tests ont été effectués avec sssd avec un domaine Active Directory et une authentification kerberos.
- Pour intégrer une machine dans le domaine Active Directory, vous pouvez choisir de suivre [cette documentation](#).
- Pour forcer la mise à jour des unités d'organisation sur l'hôte, vous pouvez appliquer un **gpupdate** à partir de la console WAPT.
- Pour que les groupes Active Directory fonctionnent correctement, vous devez vérifier que la commande **id hostname\$** renvoie la liste des groupes dont l'hôte est membre.

**Attention :** Nous avons remarqué que la requête LDAP de kerberos ne fonctionne pas si le reverse DNS record n'est pas configuré correctement pour vos contrôleurs de domaine. Ces enregistrements doivent donc être créés s'ils n'existent pas.

## 33.2.3 CentOS

### Discovery

---

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

---

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>

---

## Enterprise

Le moyen le plus sûr et le plus fiable d'installer le dernier agent WAPT sur Linux CentOS est d'utiliser le dépôt public de Tranquil IT.

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Enterprise**. Pour WAPT **Discovery** Edition, veuillez vous référer au bloc précédent.

Pour accéder aux ressources de WAPT Enterprise, vous devez utiliser le nom d'utilisateur et le mot de passe fournis par notre service commercial.

— Mise à jour de la distribution :

```
yum update
```

— Récupération de la clé .gpg :

```
wget -q -O /tmp/tranquil_it.gpg "https://wapt.tranquil.it/centos7/RPM-GPG-KEY-TISWAPT-7"; rpm --  
→import /tmp/tranquil_it.gpg
```

— Ajout du dépôt Tranquil'IT

```
cat > /etc/yum.repos.d/wapt.repo <<EOF  
[wapt]  
name=WAPT Server Repo  
baseurl=https://srvwapt-pro.tranquil.it/entreprise/centos7/wapt-2.1/  
enabled=1  
gpgcheck=1  
EOF
```

— installer l'agent WAPT en utilisant yum :

```
yum install tis-waptagent
```

## Création du fichier de configuration de l'agent

**Indication :** Utilisez l'adresse de votre serveur dans **repo\_url** et **wapt\_server**.

```
cat > /opt/wapt/wapt-get.ini <<EOF  
[global]  
repo_url=https://srvwapt.mydomain.lan/wapt  
wapt_server=https://srvwapt.mydomain.lan  
use_hostpackages=1  
use_kerberos=0  
verify_cert=0  
EOF
```

### Copie du certificat de signature du paquet

Vous devez copier manuellement, ou par script, le certificat public de votre autorité de certification de signature de paquet.

Le certificat doit se trouver sur votre machine Windows dans `C:\Program Files (x86)\wapt\ssl\`.

Copiez votre ou vos certificats dans `/opt/wapt/ssl` en utilisant **WinSCP** ou **rsync** si vous déployez sur Linux ou MacOS.

### Copie du certificat SSL/TLS

Si vous avez déjà configuré votre serveur WAPT pour utiliser les *certificats SSL/TLS avec Nginx*, vous devez copier le certificat dans votre agent WAPT Linux.

Le certificat doit se trouver sur votre ordinateur Windows, dans `C:\Program Files (x86)\wapt\ssl\server\`.

- Copiez votre ou vos certificats dans `/opt/wapt/ssl/server/` en utilisant **WinSCP** ou **rsync** si vous déployez sur Linux ou MacOS.
- Ensuite, modifiez dans votre fichier de configuration `/opt/wapt/wapt-get.ini` le chemin vers votre certificat.

---

**Indication :** Changez `YOURCERT.crt` par le nom de votre certificat.

---

```
verify_cert=/opt/wapt/ssl/server/YOURCERT.crt
```

### Enregistrer votre hôte Linux sur le serveur WAPT

- Redémarrez le service WAPT :

```
systemctl restart waptservice.service
```

- Enfin, exécutez la commande suivante pour enregistrer votre hôte Linux avec le serveur WAPT.

```
wapt-get register  
wapt-get update
```

- Redémarrez le service WAPT après l'enregistrement :

```
systemctl restart waptservice.service
```

**Félicitations**, votre agent Linux est maintenant installé et configuré et il apparaîtra maintenant dans votre console WAPT avec une icône en forme de pingouin.

### Fonctions non prises en charge

- Installation des mises à jour à l'arrêt.
- La console WAPT n'est pas actuellement disponible sur Linux.
- Une fonction spécifique à Windows.

## Particularités de la fonctionnalité du domaine

- Les tests ont été effectués avec sssd avec un domaine Active Directory et une authentification kerberos.
- Pour intégrer une machine dans le domaine Active Directory, vous pouvez choisir de suivre [cette documentation](#).
- Pour forcer la mise à jour des unités d'organisation sur l'hôte, vous pouvez appliquer un **gpupdate** à partir de la console WAPT.
- Pour que les groupes Active Directory fonctionnent correctement, vous devez vérifier que la commande **id hostname\$** renvoie la liste des groupes dont l'hôte est membre.

**Attention :** Nous avons remarqué que la requête LDAP de kerberos ne fonctionne pas si le reverse DNS record n'est pas configuré correctement pour vos contrôleurs de domaine. Ces enregistrements doivent donc être créés s'ils n'existent pas.

## 33.3 MacOS

**Attention :** Pour l'instant, l'agent n'a été testé que sur :

- High Sierra (10.13);
- Mojave (10.14);
- Catalina (10.15);
- Big Sur (10.16).

### 33.3.1 Discovery

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Discovery**. Pour WAPT **Enterprise** Edition, veuillez vous référer au bloc suivant.

**Note :** Non disponible à la date du 2024-01-09.

WAPT Discovery sera disponible plus tard. Pour la version libre, reportez-vous à la documentation de wapt-1.8 <https://www.wapt.fr/en/doc-1.8/>

### 33.3.2 Enterprise

**Important :** Suivez cette procédure pour obtenir les bons paquets pour l'édition WAPT **Enterprise**. Pour WAPT **Discovery** Edition, veuillez vous référer au bloc précédent.

Pour accéder aux ressources de WAPT Enterprise, vous devez utiliser le nom d'utilisateur et le mot de passe fournis par notre service commercial.

### En ligne de commande

---

**Indication :** Remplacez **user** et **password** dans le paramètre **baseurl** pour accéder au référentiel WAPT **Enterprise**.

---

— récupération de la liste des fichiers disponibles

```
sudo curl --user "user:password" https://srvwapt-pro.tranquil.it/entreprise/release/latest/
```

— sur le résultat, copiez `tis-waptagent-version-macos-hash.pkg`;

— créer le lien complet avec :

1. `https://srvwapt-pro.tranquil.it/entreprise/release/latest/`

et

2. `tis-waptagent-version-macos-hash.pkg` copié précédemment;

— télécharger l'agent WAPT :

**Attention :** Remplacez `<PastedLink>` par le lien créé auparavant;

```
sudo curl --user "user:password" <PastedLink> --output tis-waptagent.pkg
```

— installez le paquet téléchargé :

```
sudo installer -pkg tis-waptagent.pkg -target /
```

### Création du fichier de configuration de l'agent

---

**Indication :** Utilisez l'adresse de votre serveur dans **repo\_url** et **wapt\_server**.

---

```
sudo cat > /opt/wapt/wapt-get.ini <<EOF
[global]
repo_url=https://srvwapt.mydomain.lan/wapt
wapt_server=https://srvwapt.mydomain.lan
use_hostpackages=1
use_kerberos=0
verify_cert=0
EOF
```

## Copie du certificat de signature du paquet

Vous devez copier manuellement, ou par script, le certificat public de votre autorité de certification de signature de paquet.

Le certificat doit se trouver sur votre machine Windows dans in C:\Program Files (x86)\wapt\ssl\.

Copiez votre ou vos certificats dans /opt/wapt/ssl en utilisant **WinSCP** ou **rsync** si vous déployez sur Linux ou MacOS.

## Copie du certificat SSL/TLS

Si vous avez déjà configuré votre serveur WAPT pour utiliser les *certificats SSL/TLS avec Nginx*, vous devez copier le certificat dans votre agent WAPT Linux.

Le certificat doit se trouver sur votre ordinateur Windows, dans C:\Program Files (x86)\wapt\ssl\server\.

- Copiez votre ou vos certificats dans /opt/wapt/ssl/server/ en utilisant **WinSCP** ou **rsync** si vous déployez sur Linux ou MacOS.
- Ensuite, modifiez dans votre fichier de configuration /opt/wapt/wapt-get.ini le chemin vers votre certificat.
- Et donnez le chemin absolu de votre certificat.

```
verify_cert=/opt/wapt/ssl/server/YOURCERT.crt
```

**Indication :** Changez YOURCERT.crt par le nom de votre certificat.

## Graphiquement

- récupérer le dernier .dpkg

```
https://srvwapt-pro.tranquil.it/entreprise/release/latest/
```

- exécuter le .dpkg

### 33.3.3 Enregistrement

- redémarrez le service WAPT :

```
sudo launchctl unload /Library/LaunchDaemons/it.tranquil.waptservice.plist
sudo launchctl load /Library/LaunchDaemons/it.tranquil.waptservice.plist
```

- enfin, exécutez la commande suivante pour enregistrer votre hôte Linux avec le serveur WAPT :

```
sudo wapt-get register
```

Votre agent MacOS est maintenant installé et configuré et il apparaîtra maintenant dans votre console WAPT avec une icône de pomme.

### 33.3.4 Fonctions non prises en charge

- l'installation des mises à jour à l'arrêt ;
- La console WAPT n'est pas actuellement disponible sur MacOS ;
- une fonction spécifique à Windows ;

### 33.3.5 Particularités de la fonctionnalité du domaine

- les tests ont été effectués avec sssd avec un domaine Active Directory et une authentification kerberos ;
- pour intégrer une machine dans le domaine Active Directory, vous pouvez choisir de suivre [cette documentation](#)
- pour forcer la mise à jour des unités d'organisation sur l'hôte, vous pouvez appliquer un **gpupdate** à partir de la console WAPT ;
- pour que les groupes Active Directory fonctionnent correctement, vous devez vérifier que la commande **id hostname\$** renvoie la liste des groupes dont l'hôte est membre ;

**Attention :** Nous avons remarqué que la requête LDAP de kerberos ne fonctionne pas si le reverse DNS record n'est pas configuré correctement pour vos contrôleurs de domaine. Ces enregistrements doivent donc être créés s'ils n'existent pas.



---

## Mettre à jour les agents WAPT

---

Pour chaque *mise à jour de serveur*, vous devrez mettre à jour l'agent WAPT.

Pour cela il faut *régénérer* un agent puis le déployer.

### 34.1 Manuellement

Vous pouvez faire cela manuellement, c'est la même chose que l'*installation d'un agent*.

---

**Indication :** C'est la seule solution de mise à jour actuellement disponible pour MacOS et Linux.

---

### 34.2 Via waptupgrade

Lors de la *génération* de l'agent WAPT un paquet nommé `waptupgrade` est généré.

Ce paquet est un paquet standard de WAPT développé pour mettre à jour les agents WAPT sur les machines clientes.

---

**Indication :** Actuellement `waptupgrade` ne fonctionne que pour Windows.

---

Mettre à jour les agents WAPT en utilisant le paquet `waptupgrade` est un déroulé en deux étapes :

- tout d'abord, le paquet copie le nouveau fichier `waptagent.exe` sur le poste et crée une nouvelle tâche planifiée qui lancera le **waptagent.exe** avec un déclencheur prédéfini de 2 minutes après la création de la tâche planifiée. A ce moment-là, le paquet s'installe lui-même et l'inventaire du serveur montre l'installation du paquet en *OK*, avec la bonne version d'installée, mais l'inventaire de la console affichera toujours l'ancienne version tant que l'agent ne s'est pas encore mis à jour.

- après deux minutes la tâche planifiée démarre et exécute le **waptagent.exe**. **waptagent.exe** va couper le service WAPT local, va mettre à jour l'installation locale de WAPT puis va redémarrer le service. La tâche planifiée est alors automatiquement supprimée et l'agent WAPT va renvoyer son statut d'inventaire au serveur WAPT. Désormais, l'inventaire du serveur va afficher la nouvelle version de l'agent.

Il est recommandé d'installer waptupgrade sur tous les hôtes afin d'avoir des mises à jour automatiques pour les agents.

---

## Utiliser la console WAPT

---

Pour installer et démarrer la console WAPT, allez cette documentation pour *installer la console WAPT*.

---

**Note :** Si vous avez passé l'étape de la création de l'agent WAPT, retournez à la documentation sur *la construction de l'installateur de l'agent WAPT*.

---

Sur votre **poste de gestion**, les agents sont affichés dans la console WAPT.

---

**Note :** Si un hôte n'apparaît pas dans la console après avoir installé l'agent WAPT, ouvrez une invite de commande Windows **cmd.exe** sur l'hôte et tapez **wapt-get register**.

---

### 35.1 Ajouter un paquet à la machine

Si vous voulez ajouter un paquet directement sur la machine, il faut éditer le paquet machine.

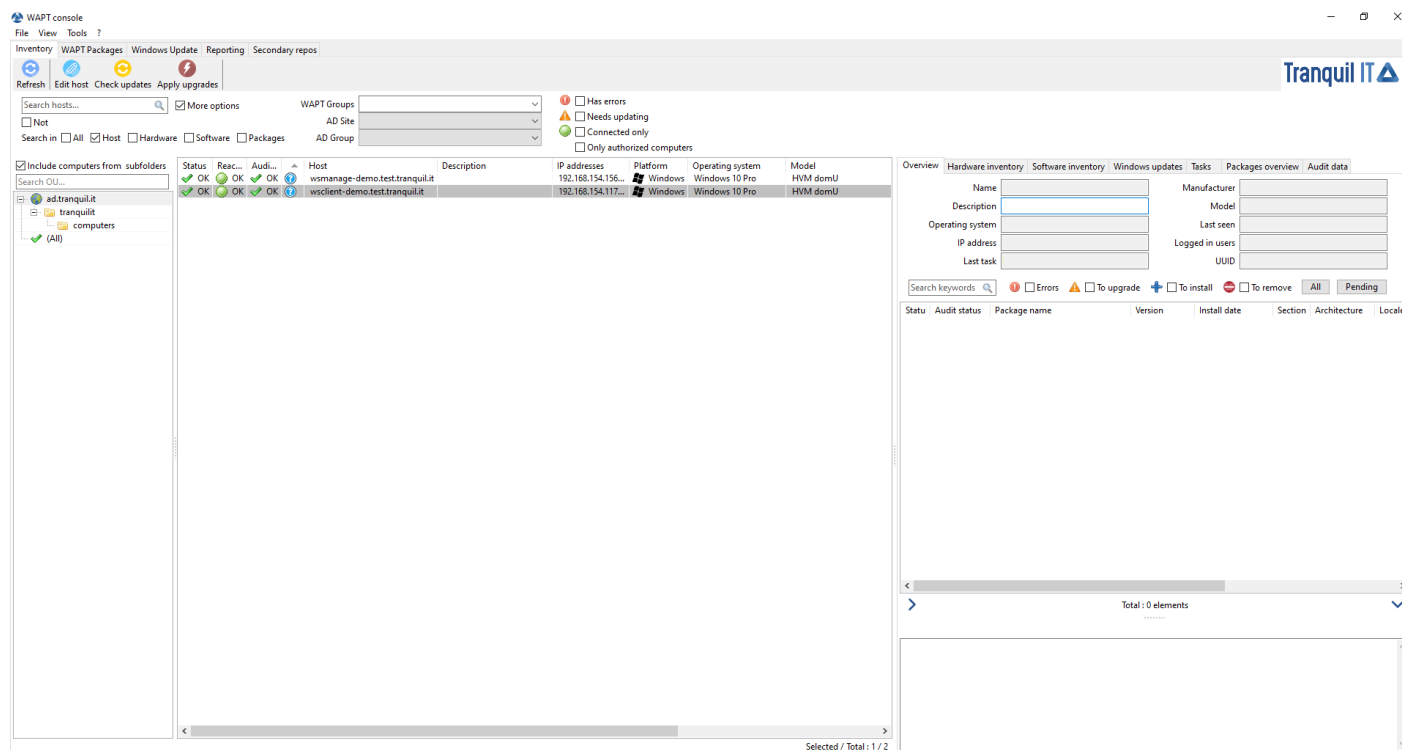
Pour cela, vous avez 3 façons :

- Double-clic sur la machine.
- Clic-droit sur la machine puis *Edit host*.
- Sélectionner une machine et utiliser le bouton *Edit host*.

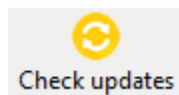
Puis, il suffit de glisser déposer le ou les paquet(s) désiré(s) et de valider.

Appuyez sur *Enregistrer* fait la même chose que faire un *update*.

Appuyer sur *Enregistrer et appliquer* revient à faire un *update* immédiatement suivi par un *upgrade*.



## 35.2 Vérifier les mises à jour sur l'hôte



Ce bouton exécute 2 actions :

1. remonter l'état actuel de l'hôte au serveur
2. le serveur indique alors si l'hôte doit récupérer des mises à jour

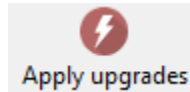
Toutes modifications de configuration nécessite un *Check updates*.

## 35.3 Appliquer les maj sur l'hôte

Ce bouton exécute les mises à jour en attente sur le poste.

**Avertissement :** A utiliser avec précaution, cela forcera la fermeture des logiciels en cours d'utilisation.

Vous pouvez utiliser à la place *Lancer les installations en attentes pour les applications non lancées* pour éviter toute perte de travail



## 35.4 Effectuer une recherche globale sur tous les hôtes

Effectuer des recherches globales avec tous les critères présentés ci-dessous est possible.

Choisir les filtres à cocher ou décocher.

The screenshot shows the WAPT interface with the following elements:

- Navigation tabs: Inventory, WAPT Packages, Windows Update, Reporting, Secondary repos.
- Action buttons: Refresh, Edit host, Check updates, Apply upgrades.
- Search bar: Search hosts... with a search icon.
- More options:  More options.
- WAPT Groups: A dropdown menu.
- AD Site: A dropdown menu.
- AD Group: A dropdown menu.
- Search filters:
  - Not
  - Search in:  All,  Host,  Hardware,  Software,  Packages
- Filter checkboxes:
  - Has errors
  - Needs updating
  - Connected only
  - Only authorized computers

TABLEAU 1 – Choix des filtres

Options possibles	Description
<i>Machine</i>	La section <i>Host</i> dans l'onglet <i>Inventaire matériel</i> quand un hôte est sélectionné
<i>Matériel</i>	La section <i>DMI</i> dans l'onglet <i>Inventaire matériel</i> lorsqu'un hôte est sélectionné
<i>Logiciel</i>	La section <i>Inventaire logiciel</i> lorsqu'un hôte est sélectionné
<i>Paquets</i>	Liste des paquets installés sur les hôtes sélectionnés
<i>Has errors</i>	Rechercher uniquement les hôtes dont les tâches ne se sont pas correctement terminées
<i>Needs updating</i>	Rechercher uniquement les hôte nécessitant une mise à jour
<i>Connectés seulement</i>	Rechercher uniquement les hôtes connectés
<i>Seuls les ordinateurs autorisés</i>	Rechercher uniquement les hôtes autorisés par le certificat de l'utilisateur en cours
<i>WAPT Group</i>	Filter les hôtes basés sur leur appartenance / dépendance à un paquet groupe WAPT
<i>AD Site</i>	Filter les hôtes basés sur leur appartenance / dépendance à Site et Services Active Directory
<i>AD Group</i>	Filter les hôtes basés sur leur appartenance / dépendance à un groupe Active Directory

**Indication :** Les filtres fonctionnent avec des expressions régulières.

## 35.5 Afficher l'inventaire

Lorsque les agents WAPT s'enregistrent avec un **register**, ils envoient des informations au serveur WAPT.

Les informations affichées dans la console ne sont pas mises à jour en temps réel, vous devez rafraîchir l'affichage pour voir les nouveaux statuts et informations.

Cliquez sur le bouton *Refresh* ou appuyez sur F5 sur le clavier.

Status	Reac...	Audi...	...	Host	Description	IP addresses	Platform	Operating system	Model
✓ OK	OK	OK	?	wsmanage-demo.test.tranquil.it		192.168.154.156...	Windows	Windows 10 Pro	HVM domU
⚠ T...	OK	OK	?	wsclient-demo.test.tranquil.it		192.168.154.117...	Windows	Windows 10 Pro	HVM domU

La console WAPT liste les hôtes qui sont enregistrés sur le serveur WAPT ainsi que des informations utiles pour gérer les hôtes. Sélectionner un hôte affiche ses informations sur le panneau de droite de la console WAPT (*Hardware inventory* et *Software inventory*). Plus de détails *ici*.

## 35.6 Comment effectuer des actions sur les hôtes ?

Certaines actions ne sont pas présente en multi sélection, cf :

TABLEAU 2: Liste des actions disponibles qui peuvent être faites sur les hôtes dans la console WAPT

Nom	Multi-sélection
Edite hôte	<b>non</b>
Vérifier les mises a jour	<b>oui</b>
Appliquer les mises à jour	<b>oui</b>
Appliquer les mises a jours d'applications qui ne sont pas lancées	<b>oui</b>
Proposer la mise a jours a l'utilisateur	<b>oui</b>
Envoyer un message aux utilisateurs	<b>oui</b>
Lance les audits des paquets	<b>oui</b>
Ajouter un paquet aux dépendances de la machine	<b>oui</b>
Retire un paquets des dépendances de l'hôte	<b>oui</b>
Re-signe les paquets <i>hôtes</i>	<b>oui</b>
Ajouter un paquet dans les conflits de la machine	<b>oui</b>
Retire un paquet des conflits de l'hôte	<b>oui</b>
Supprime le poste	<b>oui</b>
Connexion en RDP	<b>non</b>
Assistance a distance	<b>non</b>
Mesh remote desktop	<b>oui</b>
Gestion de l'ordinateur windows	
Mettre a jour les GPO sur l'hôte	<b>oui</b>
Lance CleanMgr on host	<b>non</b>
Gestion de l'ordinateur	<b>non</b>
Gérer les utilisateurs et groupes	<b>non</b>
Gestion des services	<b>non</b>
Allumer avec WakeOnLan	<b>oui</b>
Redémarre les ordinateurs	<b>non</b>
Eteints les ordinateurs	<b>non</b>
Déclenche le scan des mises a jours Windows manquantes	<b>oui</b>
Déclenche le téléchargement des mises a jours Windows manquantes	<b>oui</b>
Déclenche l'installation des mises a jours Windows manquantes	<b>oui</b>
Rafraichir l'inventaire	<b>oui</b>
Lance un redémarrage de waptservice	<b>oui</b>

**Note :** Pour les détails de ces actions, référez-vous à *Actions détaillées sur l'hôte*.

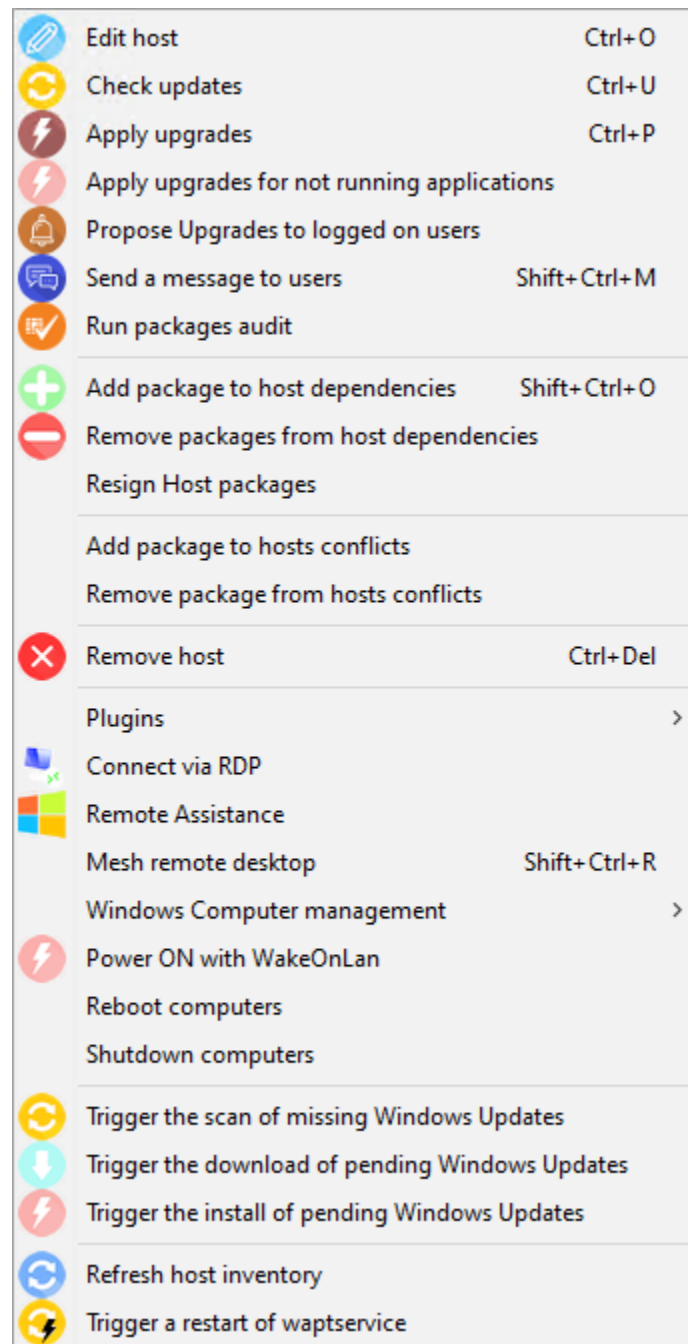


FIG. 1 – Menu de configuration de l'hôte

## 35.7 Importe des paquets depuis un dépôt externe

### 35.7.1 Principe d'importation des paquets

Importer un paquet WAPT consiste à :

- Importer un paquet WAPT existant depuis un dépôt externe.
- Changer ses préfixes (par exemple de *tis* à *test*).
- Re-signer le paquet WAPT avec la clé privée *Administrateur* pour permettre de déployer des paquets dupliqués sur vos postes équipés de l'agent WAPT.
- Enfin, le téléverser sur le dépôt WAPT principal.

**Attention :** En important un paquet dans votre dépôt et en le signant, **VOUS DEVENEZ ALORS RESPONSABLE** de ce paquet et de ce qu'il fait. **Il a été signé avec votre propre clé privée.**

**Tranquil IT** déclare se dégager de toutes responsabilités si vous choisissez d'utiliser des paquets WAPT venant de ses dépôts. Sans un contrat de support, Tranquil IT ne garantit pas la pertinence du paquet pour vos propres cas d'usage, et ne garantit pas non plus la capacité du paquet à convenir à la politique de sécurité interne à votre *Organisation*.

- Allez dans l'onglet *Private repository*.

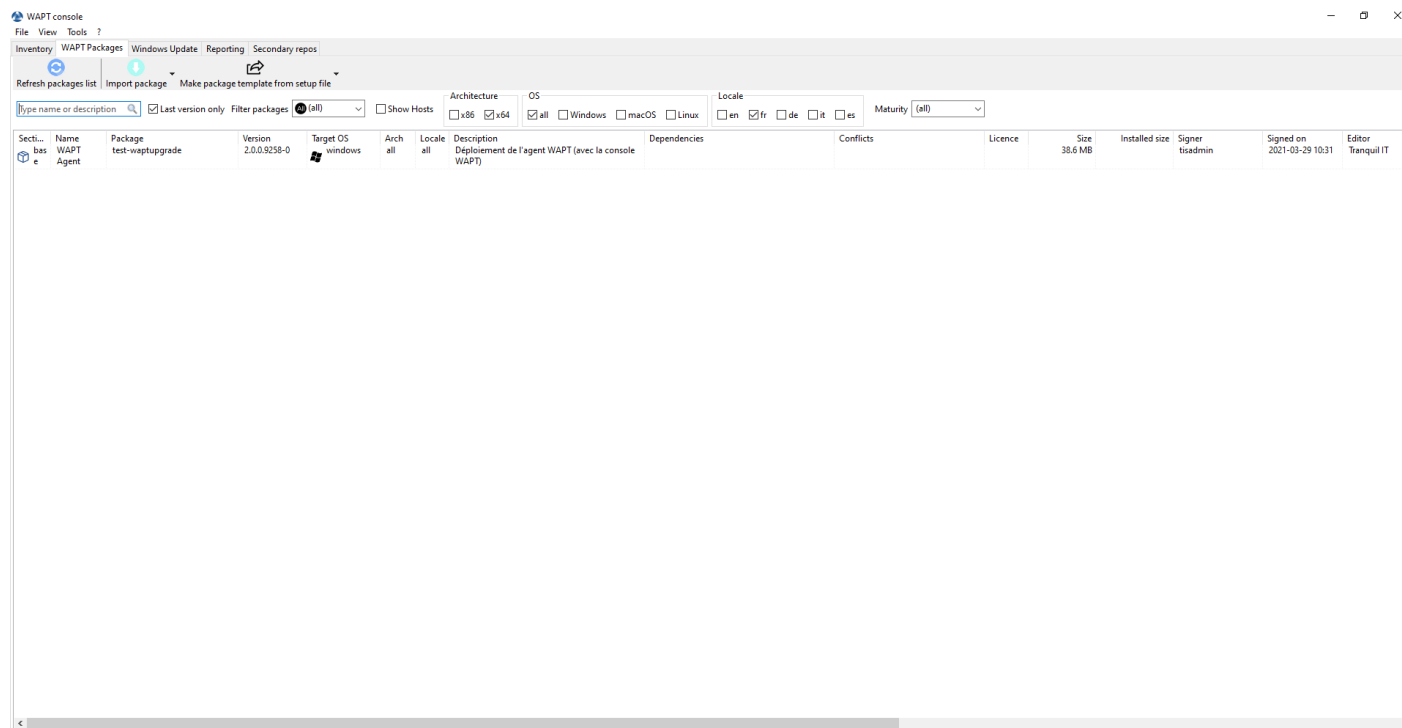


FIG. 2 – Les logiciels disponibles sont affichés dans la console WAPT

Chaque version du paquet logiciel disponible est affichée dans le dépôt.

Si aucun paquet n'a été importé, la liste est vide. Seul le paquet « *test-waptupgrade* » sera présent si l'agent WAPT a été généré précédemment. Visitez la documentation sur *la création d'un agent WAPT*.

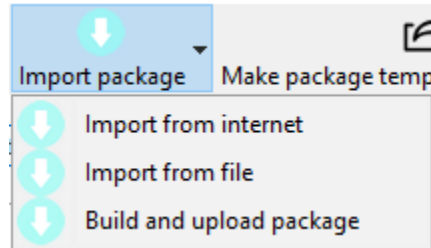


## 35.7.2 Importer un paquet depuis un dépôt externe sur Internet

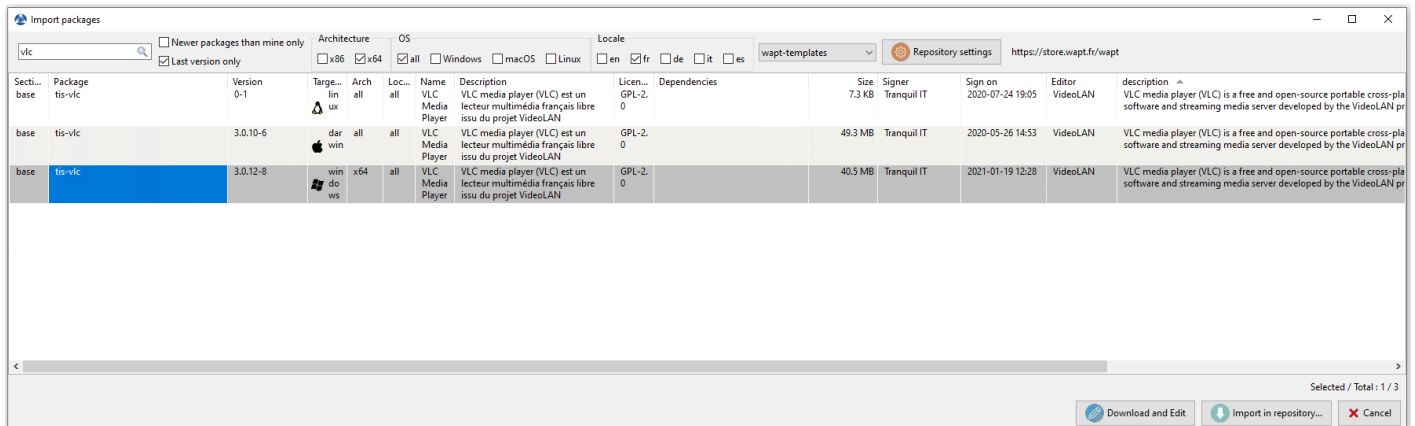
Cette première méthode vous permet de télécharger des paquets directement depuis un dépôt externe WAPT dans votre *Organisation*. Par défaut le dépôt Tranquil'IT est présent, pour ajouter un autre dépôt cf *paramètres de configuration des dépôts externes* ;

**Note :** Par défaut, les certificats TLS et SSL des dépôts externes sont vérifiés.

— Cliquez sur *Importer un paquet* et *Importer depuis Internet*.



**Note :** La grille d'affichage montre la liste de paquets disponibles sur le dépôt distant. Il est possible de choisir la plateforme, l'OS et la langue.



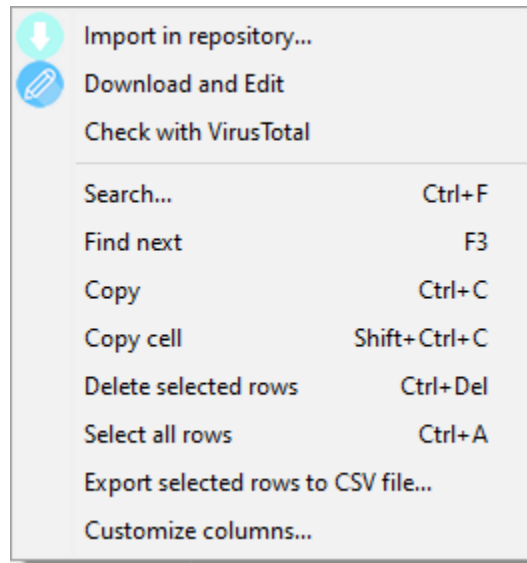
- Il y a 2 méthodes pour importer un paquet WAPT :
  - pour importer un paquet, sélectionner un paquet puis *Clic-droit* → *Importer* ;
  - ou en bas à droite de la console *Importer dans le dépôt* ;
- Valider l'import dans votre dépôt local.

**Note :** Il est possible de *changer la maturité d'un paquet WAPT* avant d'importer le paquet dans votre dépôt privé.

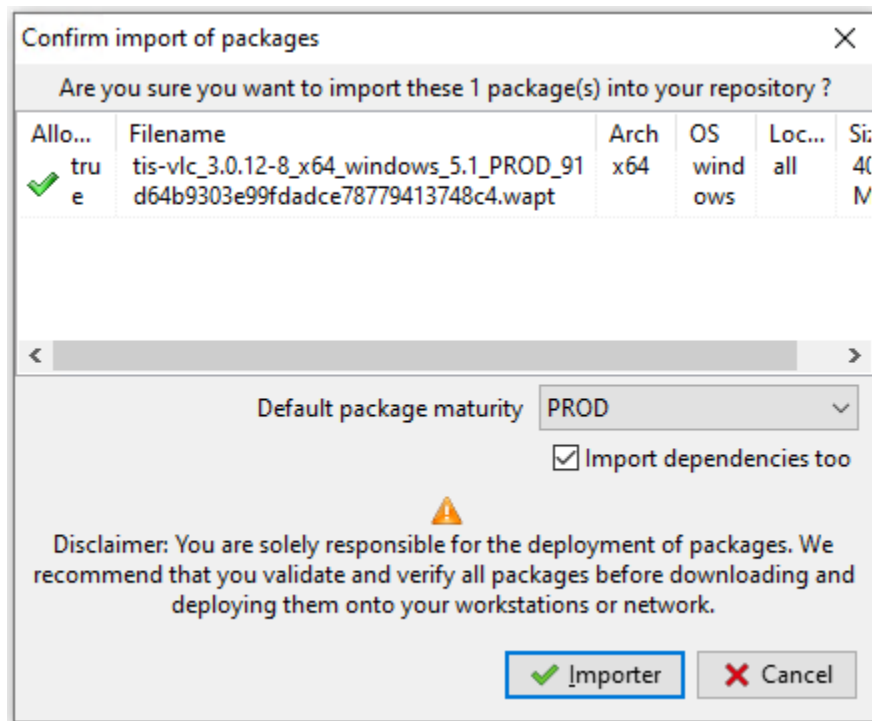
- Le téléchargement du paquet commence.
- Puis, entrer le mot de passe de votre clé privée.

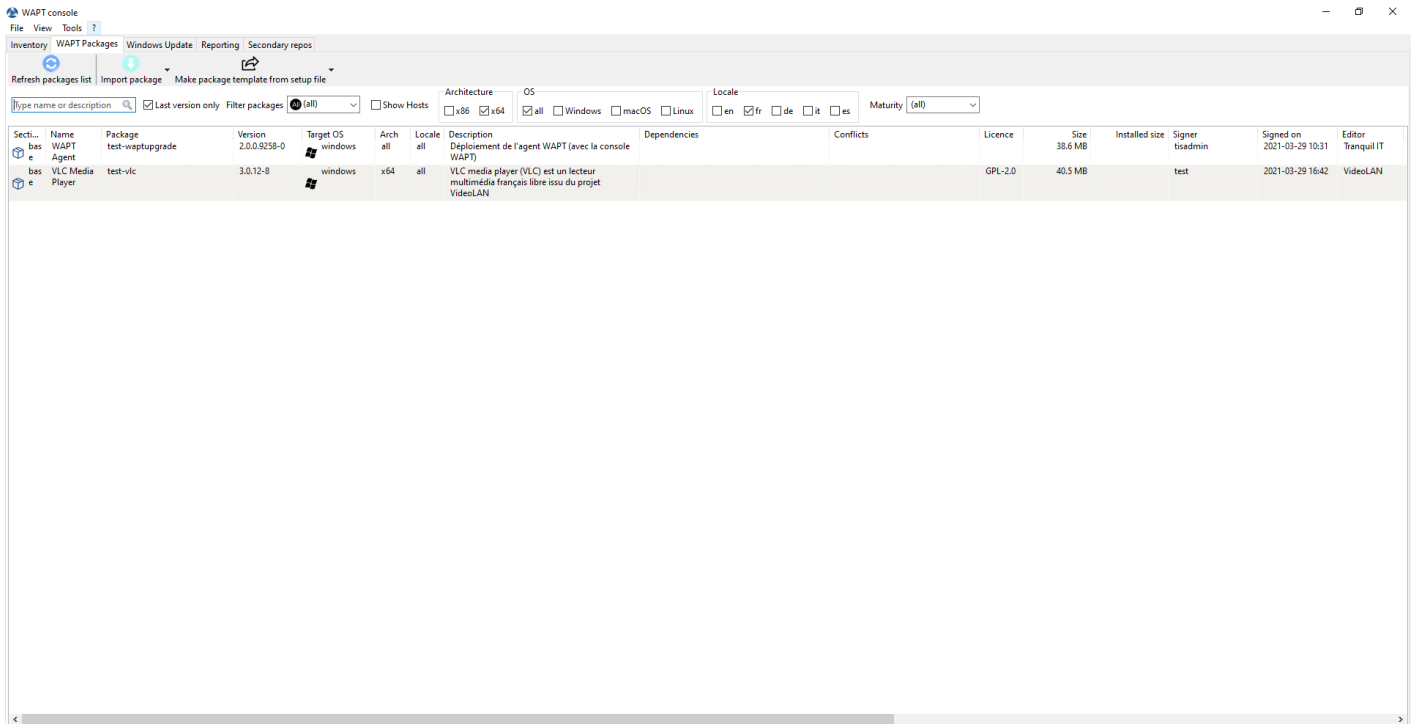
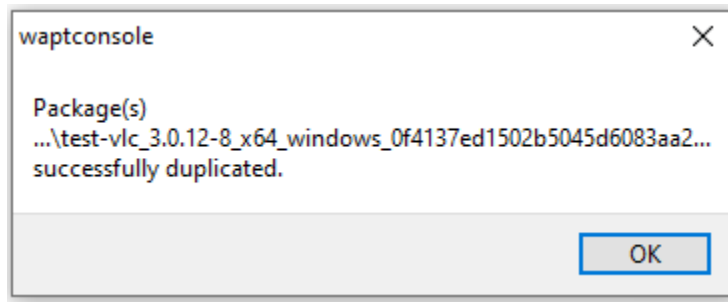
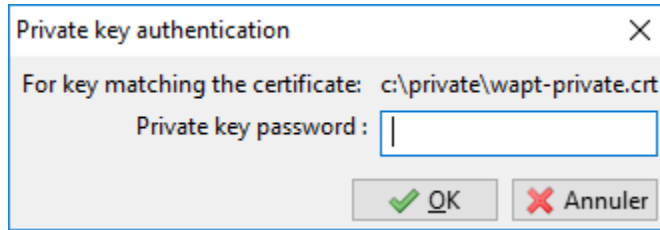
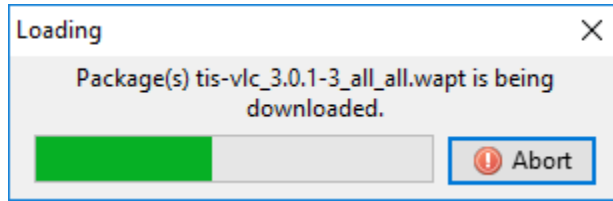
La console WAPT confirme que le paquet a bien été dupliqué sur votre dépôt WAPT local.

Le paquet apparaît alors sur votre dépôt local WAPT avec le préfixe de votre Organisation.



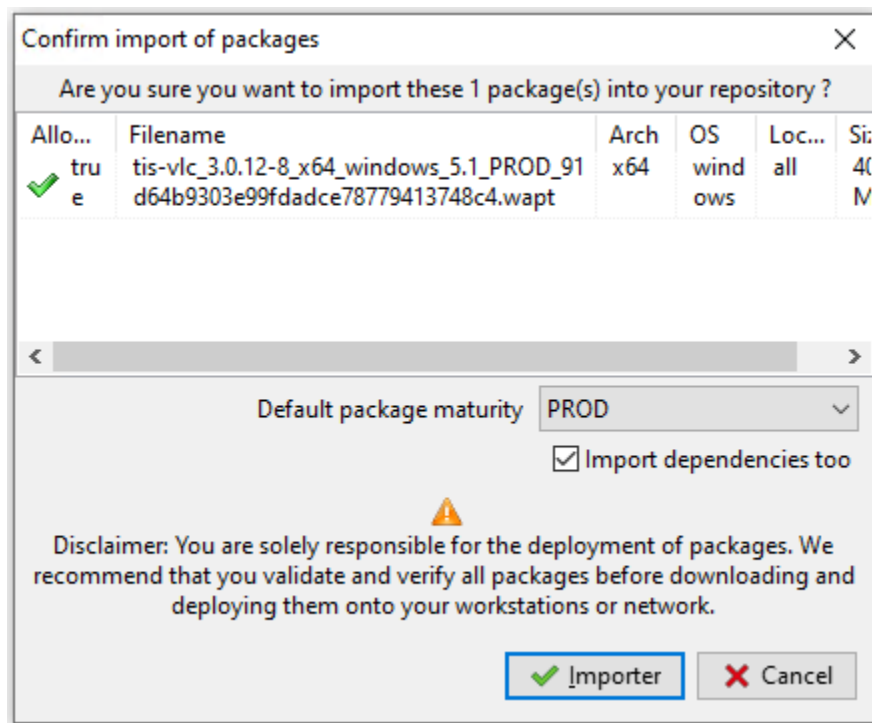
wapt-resources/wapt\_console-import-package-options-2.png





## Changer la maturité d'un paquet WAPT avant de l'importer dans le dépôt

Il est possible de changer la maturité d'un paquet WAPT avant de le charger dans votre dépôt privé en choisissant **DEV**, **PREPROD** ou **PROD** dans *Maturité des paquets par défaut*.



## Editez un paquet avant de l'importer

Il est possible d'éditer un paquet téléchargé depuis un dépôt externe avant de l'importer dans dépôt WAPT principal.

- Pour cela 2 choix disponible :
  - pour importer un paquet, sélectionner un paquet puis *Clic-droit* → *Importer* ;
  - ou en cliquant sur *Télécharger et éditer* en bas à droite de la fenêtre ;

**PyScripter**, s'il est installé, va ouvrir les fichiers `control` et `setup.py` du paquet WAPT.

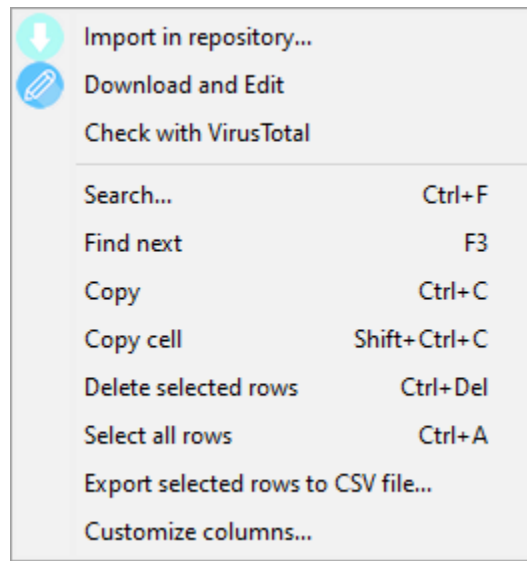
Pour plus d'information, visitez la documentation sur *créer un paquet WAPT*.

## 35.8 Dupliquer des paquets depuis un dépôt externe

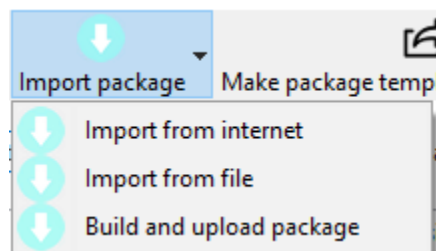
### 35.8.1 Importer un paquet WAPT depuis un fichier

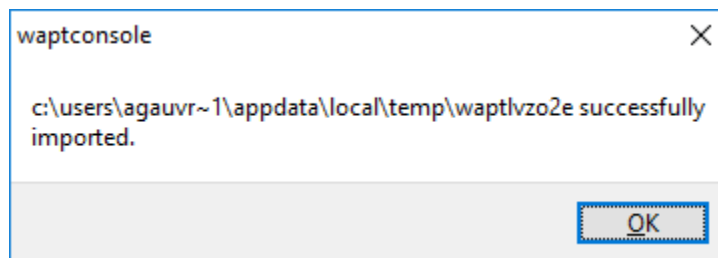
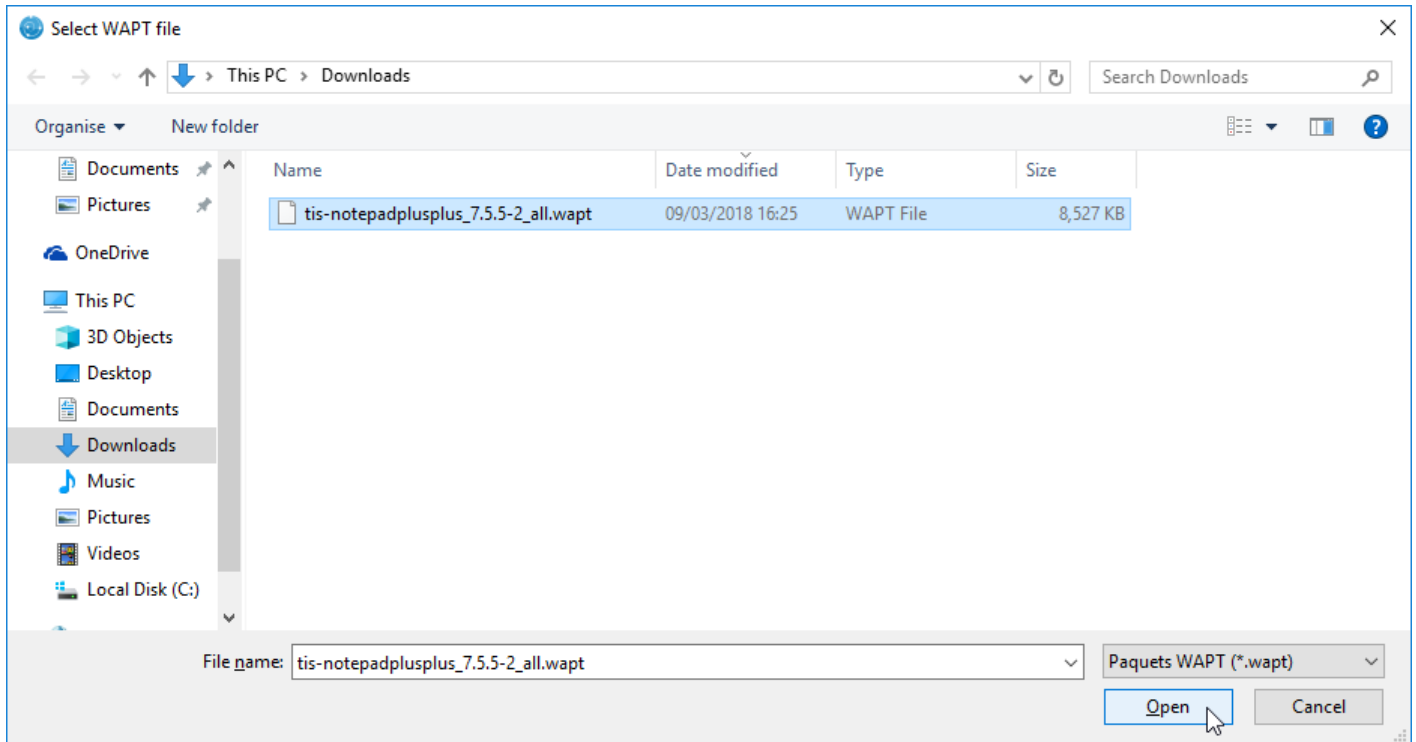
Vous pouvez importer un fichier `.wapt` depuis n'importe quel stockage.

- Cliquer sur *Importer un paquet* puis sur *Importer depuis un fichier*.
- Sélectionner le fichier à importer.
- Cliquez sur *Ouvrir* pour importer le fichier.



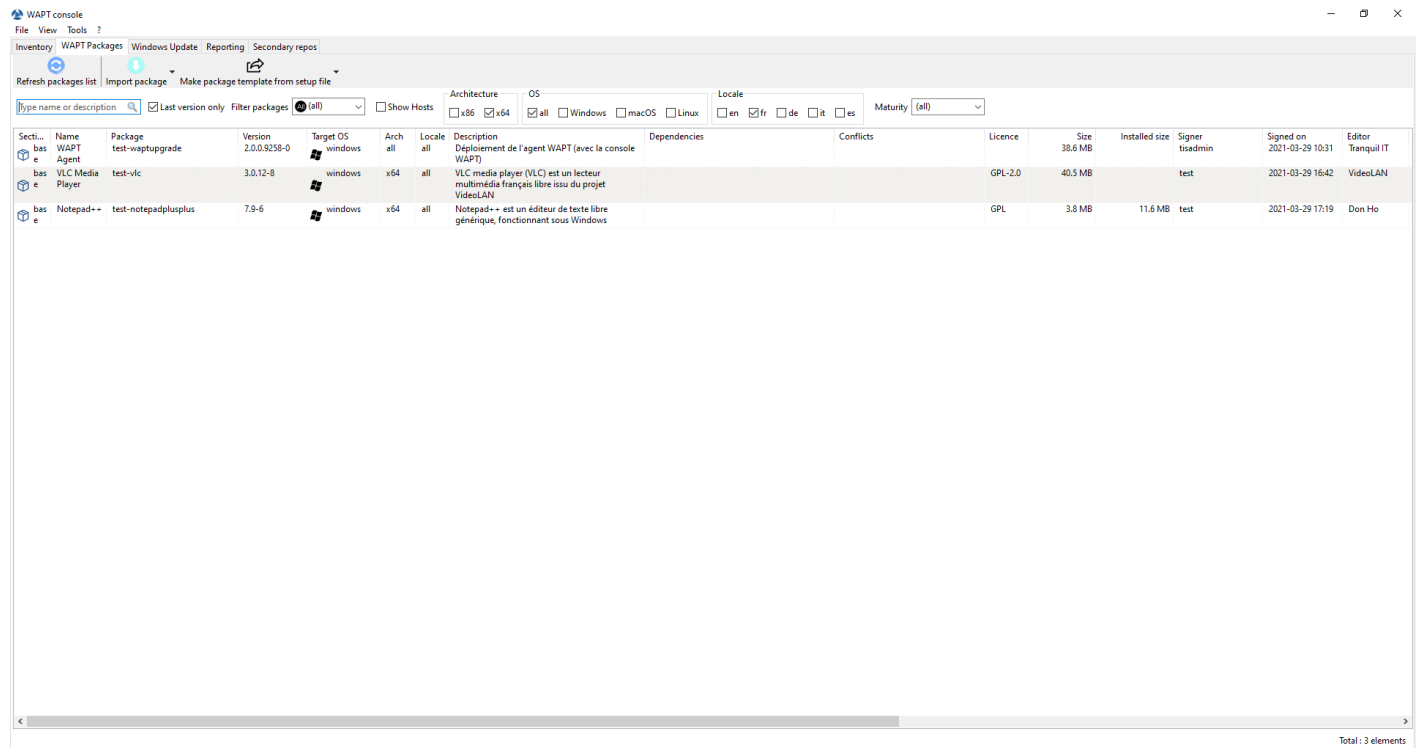
wapt-resources/wapt\_console-import-package-options-2.png





La console WAPT confirme que le paquet a bien été dupliqué sur votre dépôt WAPT local.

La paquet apparaît alors sur votre dépôt local WAPT avec le préfixe de votre Organisation.



**Note :** Il n'est pas possible de changer la maturité avant d'importer dans ce cas.

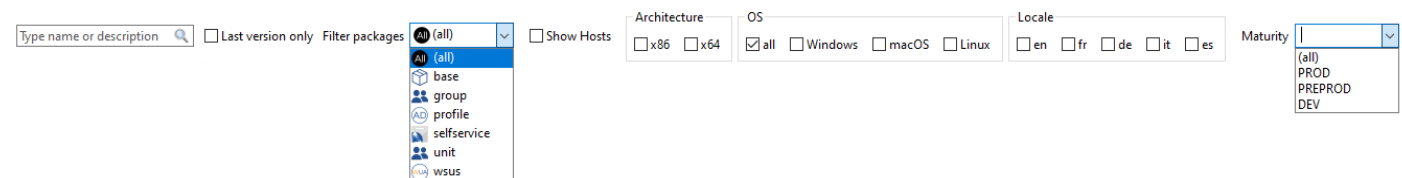
## 35.8.2 Changer le préfixe et resigner un paquet WAPT

Lors de l'import de votre nouveau paquet WAPT dans votre dépôt privé, le changement du préfixe et la re-signature du paquet WAPT sont transparents et automatique.

## 35.9 Gérer les paquets sur le dépôt

Dans l'onglet *WAPT Packages*, la liste des paquets actuellement disponibles sur le dépôt WAPT apparaît. Par défaut, la console va uniquement montrer la dernière version des paquets.

Un barre de recherche est aussi disponible pour filtrer les paquets. *Il est possible de spécifier un filtre.*



### 35.9.1 Afficher tous les paquets

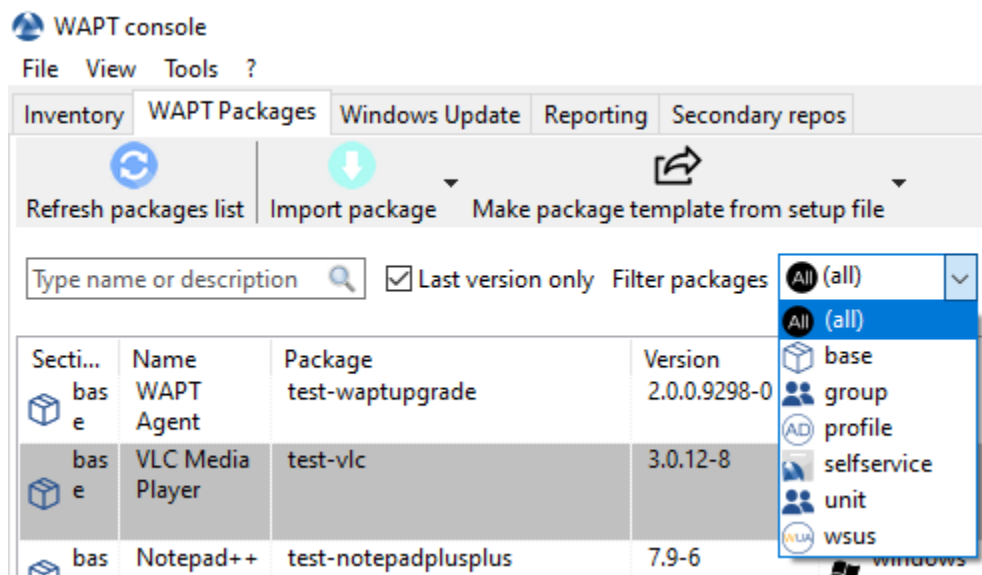
Pour afficher toutes les versions des paquets, décochez *Dernière version seulement*.

### 35.9.2 Filtrer par type de paquet

Pour afficher un type de paquet spécifique, utilisez *Filter packages* :

Les types de paquet sont :

- *all*;
- *base*;
- *group*;
- *profile*;
- *selfservice*;
- *unit*;
- *waptwua*;



### 35.9.3 Autres filtres

Architecture:  x86  x64

OS:  all  Windows  macOS  Linux

Locale:  en  fr  de  it  es

Maturity:

Les autres filtres disponibles sont :

- architecture :
  - *x86*;
  - *x64*;
- OS (Operating System) :
  - *all*;
  - *Windows*;
  - *macOS*;



- *Linux* ;
- locale :
  - en ;
  - fr ;
  - de ;
  - it ;
  - es ;
- maturité :
  - PROD ;
  - PREPROD ;
  - DEV ;

### 35.9.4 Faire une recherche basé sur un paquet WAPT

Dans l'onglet le dépôt, sélectionnez le paquet puis cliquez sur *Afficher les clients*.

La grille va afficher les hôtes sur lesquels le paquet est installé. Notez que le filtre n'est actif que sur l'attribut *Package* du paquet sélectionné.

Les différentes colonnes affichent des informations à propos des paquets installés sur la machine (e.g *package version*, *package status*, *audit status*, *installation date*, *architecture*).

The screenshot shows the WAPT console interface. At the top, there are tabs for 'Inventory', 'WAPT Packages', 'Windows Update', 'Reporting', and 'Secondary repos'. Below the tabs, there are several filters and search options. A search bar contains 'vlc'. The 'Architecture' filter is set to 'x64', and the 'OS' filter is set to 'all'. The 'Locale' filter is set to 'fr'. The 'Maturity' filter is set to 'all'. Below the filters, there is a table with the following columns: Secti..., Name, Package, Version, Target OS, Arch, Locale, Description, Dependencies, Conflicts, Licence, and Size. The table contains one row for the 'vlc' package. Below the table, there is a section for 'Hosts' with a search bar and a 'Refresh' button. The 'Hosts' section contains a table with the following columns: Host reach..., Computer name, Host description, Status, Audit st..., Package, Version, Package description, Install date, Arch, Loc..., Matur..., and Connecte... The table contains one row for the host 'wsclient-demo.te...'. The status is 'OK', the audit status is 'OK', the package is 'test-vlc', the version is '3.0.12-8', the package description is 'VLC media player (V...)', the install date is '2021-03-30 14:31', the architecture is 'x64', the locale is 'all', the maturity is 'PROD', and the connection is '192.168.1...'.

Secti...	Name	Package	Version	Target OS	Arch	Locale	Description	Dependencies	Conflicts	Licence	Size
bas e	VLC Media Player	test-vlc	3.0.12-8	windows	x64	all	VLC media player (VLC) est un lecteur multimédia français libre issu du projet VideoLAN			GPL-2.0	40.5 MB

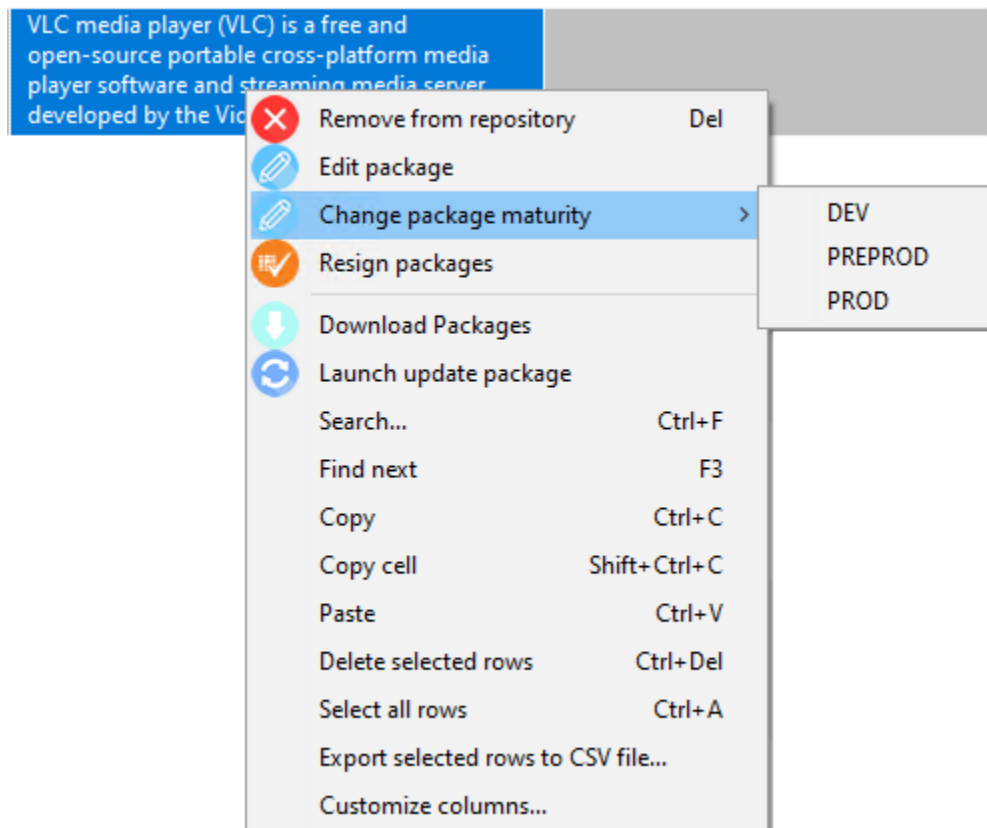
  

Host reach...	Computer name	Host description	Status	Audit st...	Package	Version	Package description	Install date	Arch	Loc...	Matur...	Connecte...
OK	wsclient-demo.te...		OK	OK	test-vlc	3.0.12-8	VLC media player (V...	2021-03-30 14:31	x64	all	PROD	192.168.1...

Vous pouvez aussi ajouter les colonnes *Log install* et *Last Audit Output* pour jeter un coup d'oeil aux journaux d'installation et aux journaux d'audit.

### 35.9.5 Changer la maturité d'un paquet WAPT après l'import sur le dépôt

Lorsqu'un paquet est importé sur le dépôt WAPT il est possible de changer la maturité en faisant un clic-droit sur le paquet WAPT. Sélectionnez votre maturité sur le menu *Change packages maturity*.




Une fois la maturité choisie une fenêtre apparait :

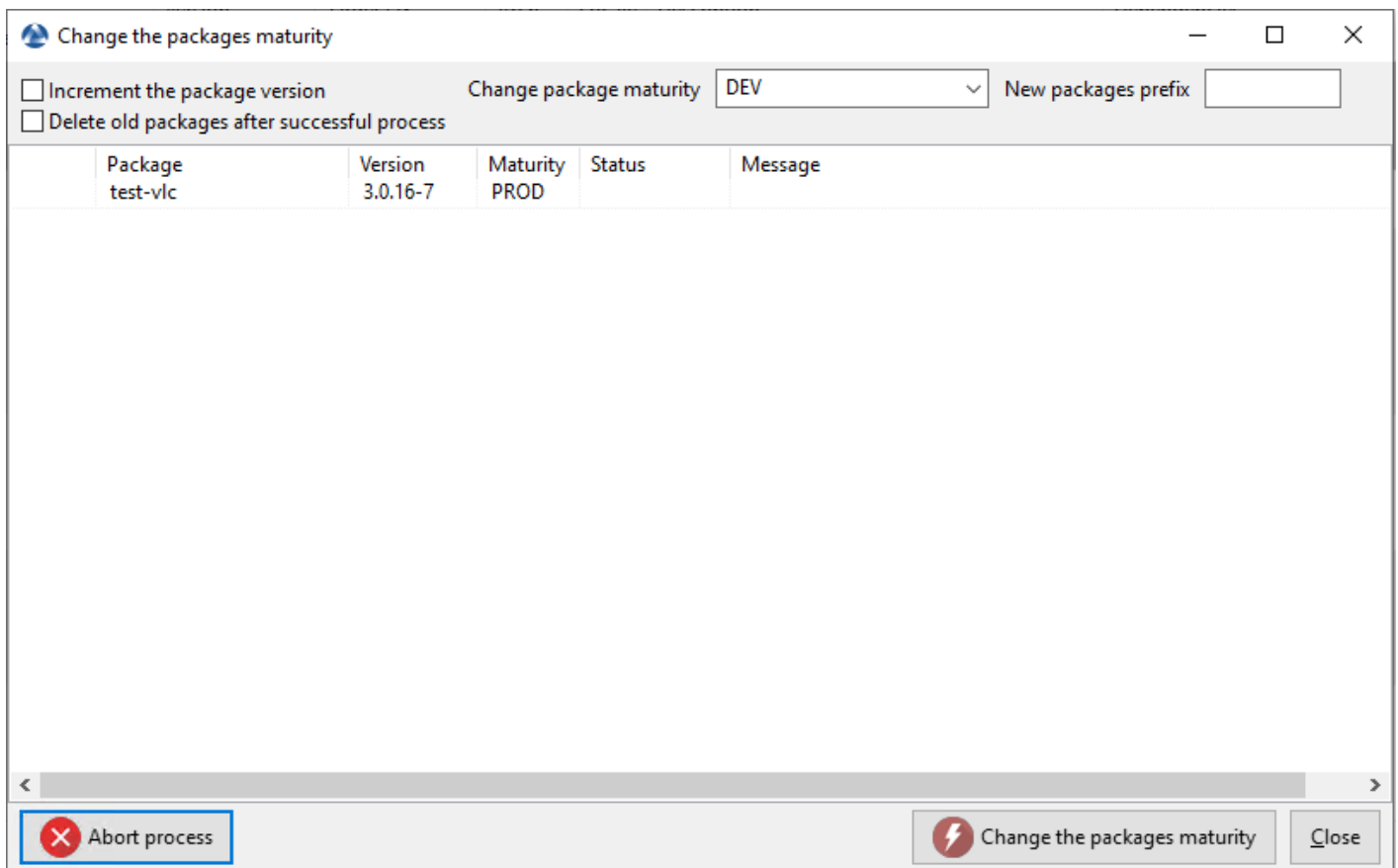
Plusieurs choix sont possibles :

Label	Description
<i>Package 'Increment the package version'</i>	Incrémenter la version du paquet (numéro de version après -).
<i>Delete old packages after successful process</i>	Supprimer les vieux paquets WAPT après avoir changé la maturité.
<i>Change la maturité du paquet</i>	Configurer la nouvelle maturité du paquet WAPT.
<i>Prefix du nouveau paquet</i>	Configurer un nouveau préfixe pour les paquets WAPT.

**Note :** Vous pouvez arrêter le processus en cliquant le bouton *Arrêter le processus*.

Une fois fini, le statut passe en .

**Indication :** Vous pouvez changer la maturité de multiples paquets en une fois



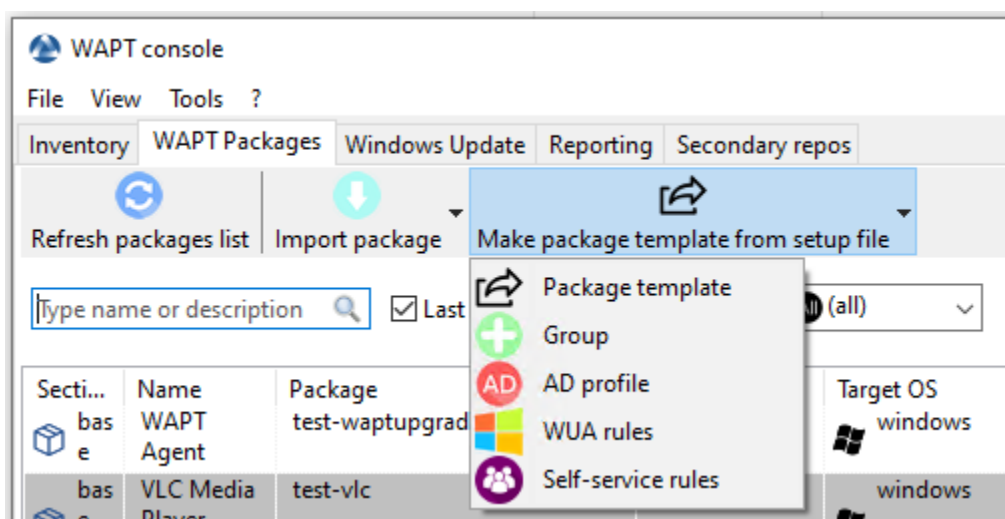
**Avertissement :** Le changement de maturité du paquet va modifier le hash du paquet.

Si le paquet est utilisé dans une GPO, comme **waptupgrade**, vous devrez changer le hash de votre GPO.

### 35.9.6 Créer un groupe de paquet

Les groupes de paquet vous permettent de créer un paquet contenant d'autres paquets qui seront affectés en tant que dépendance à un hôte.

Pour créer un groupe de paquet, aller l'onglet *Dépôt privé*, puis cliquer sur *Générer un modèle de paquet* et enfin sélectionnez *Groupe*.



- Change le nom dans *nom du paquet*.
- Remplissez la description.
- Ajoutez des paquets au groupe en les glissant et déposant ou bien en faisant un clic-droit sur le nom du paquet, et ajoutez-le au paquet.
- Cliquer sur *Enregistrer* pour enregistrer le paquet groupe.

---

**Indication :** Pour désinstaller un paquet, il est possible de l'ajouter comme interdit à un groupe de paquet. Le paquet interdit, si installé, sera supprimé avant que d'autres paquets soient installés.

---

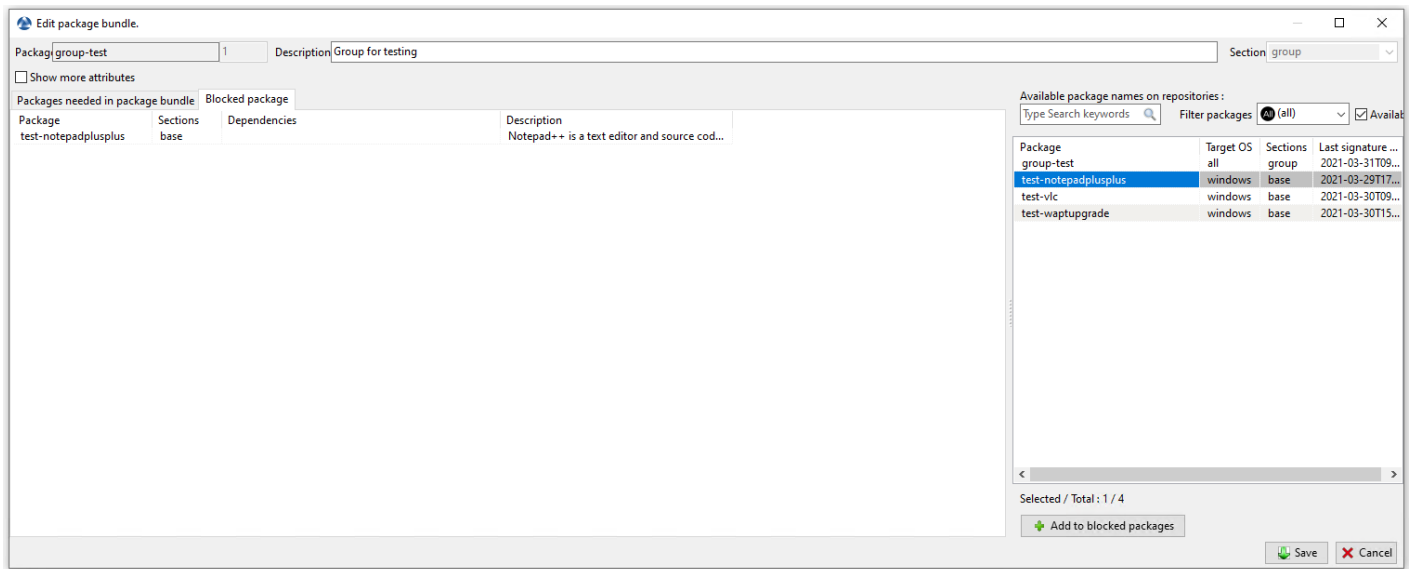
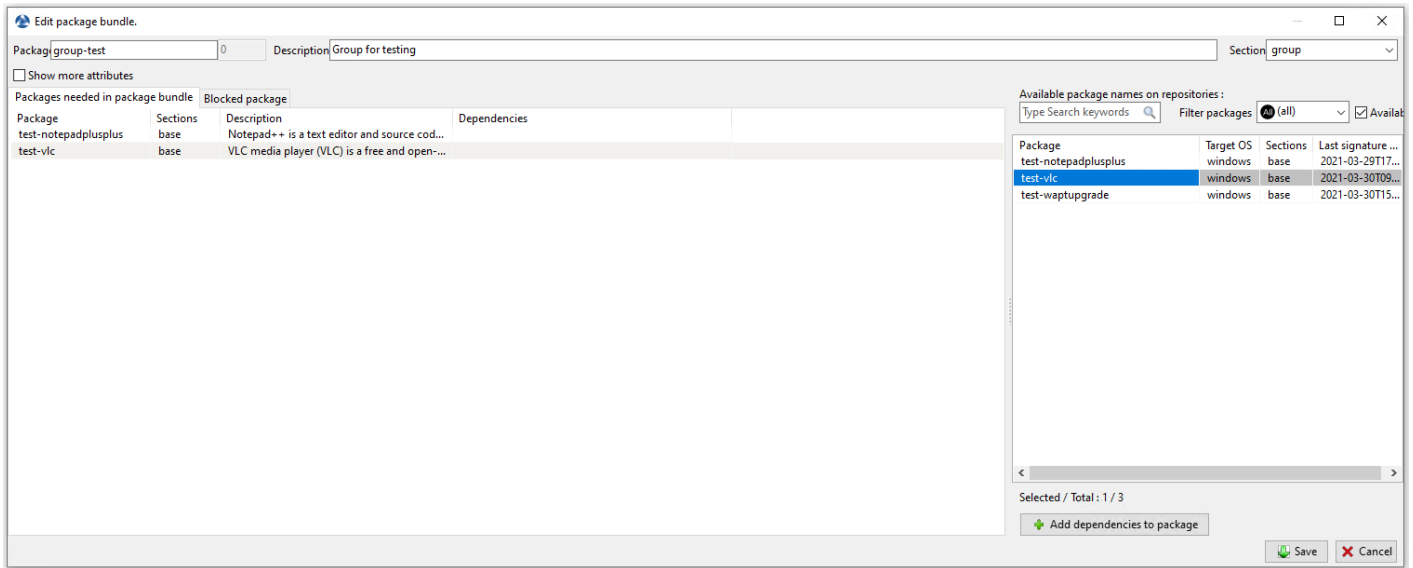
### 35.9.7 Supprimer un paquet WAPT

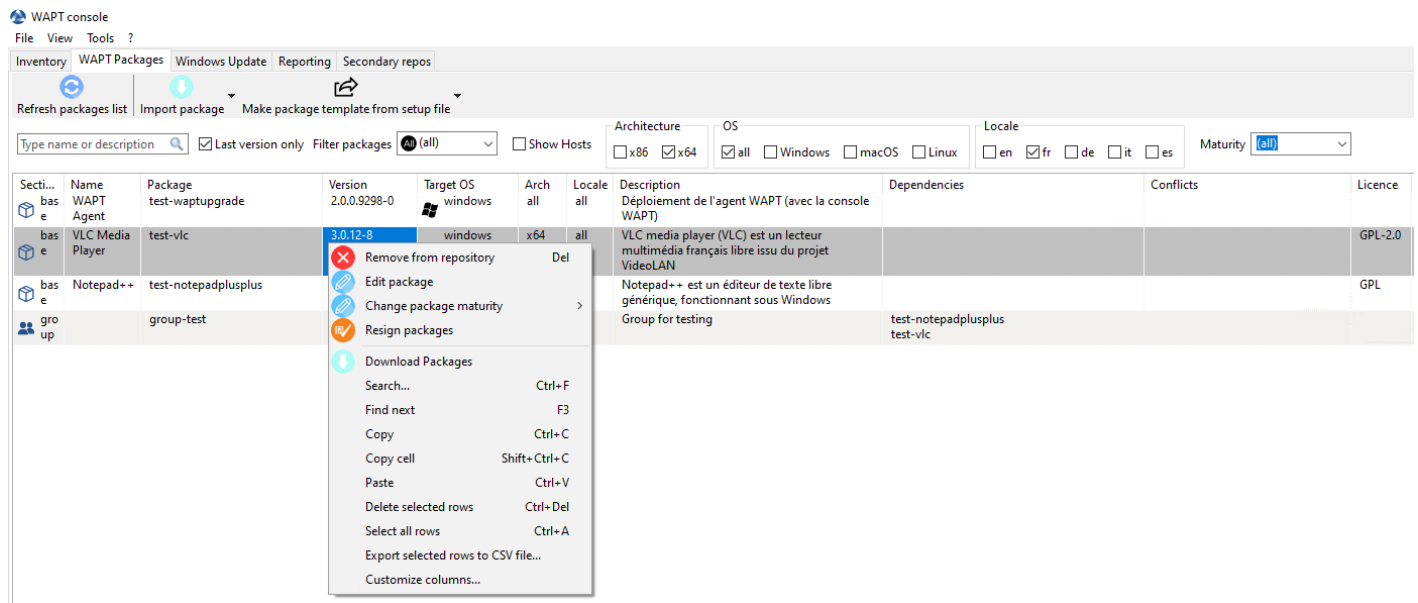
Pour supprimer un paquet depuis le dépôt, *Clic-droit* → *Supprimer du dépôt*.

---

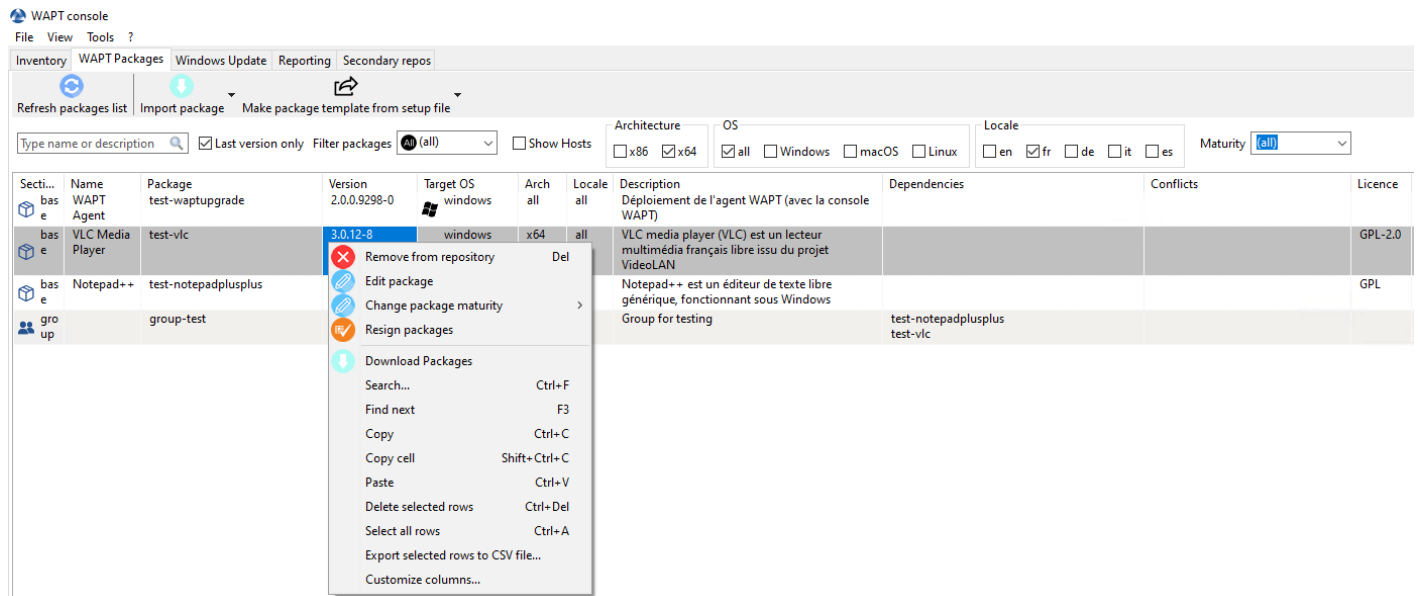
**Indication :** Vous pouvez sélectionner plusieurs paquets à la fois pour les supprimer.

---





## 35.9.8 Editer un paquet WAPT



Pour éditer un paquet, *clique-droit* → *Editer le paquet*.

Le paquet sera téléchargé localement dans **le répertoire de développement du paquet de base** défini dans les *paramètres de la console*.

Si **PyScripter** est installé, il ouvrira automatiquement les fichiers `control` et `setup.py`.

Une fois édité vous pouvez téléverser *depuis la console wapt*.

### 35.9.9 Déployer des paquets WAPT puis la console WAPT

Vous pouvez déployer des paquets de multiples façons :

- Directement en *ajoutant un paquet a le ou les hôte(s) sélectionné(s)*.
- En *ajoutant un paquet WAPT à une Unité Organisationnelle* dont l'hôte est membre.
- En *ajoutant un paquet à un profile d'hôte* qui est appliqué à l'hôte.
- En *ajoutant le paquet à un paquet groupe* dont l'hôte est membre.





---

## Utiliser la console WAPT avancé

---

Cette page détaille l'utilisation avancée de la console WAPT.

### 36.1 Utiliser des paquets profile dans WAPT

#### 36.1.1 Principe de fonctionnement

WAPT Enterprise propose une fonctionnalité paquet profile Active Directory.

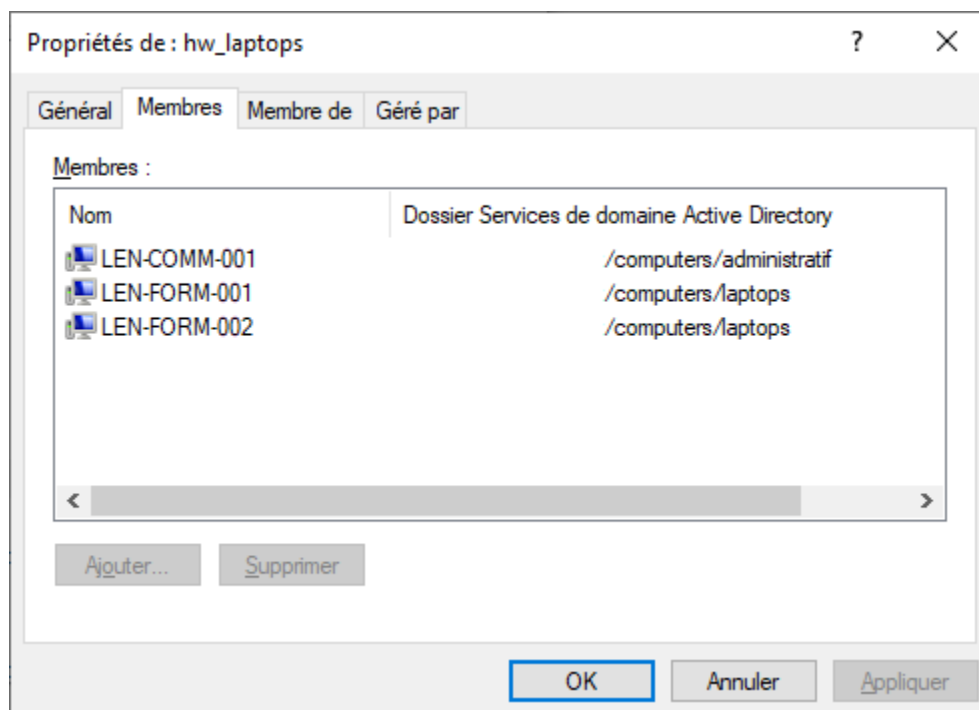
Cela automatise l'installation du logiciel WAPT ainsi que ses paquets de configuration sur des hôtes, basé sur leur appartenance aux Groupes de Sécurité Ordinateur Active Directory.

---

**Important : Les groupes de Sécurité Ordinateur Active Directory contiennent des Ordinateurs et non pas des Utilisateurs.**

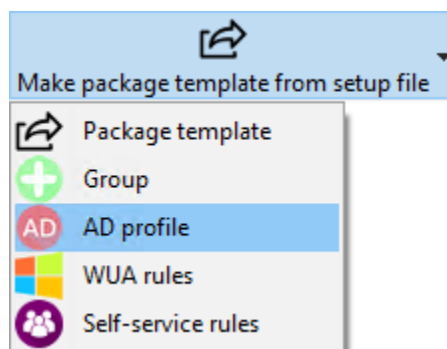
---

**Avertissement :** L'installation automatique de logiciels et de configurations en fonction de l'utilisateur et de l'appartenance à un groupe d'utilisateurs n'est pas implémentée avec WAPT et une telle implémentation n'est pas souhaitable. Le cas d'utilisation de l'installation de logiciels en fonction du profil de l'utilisateur est mieux servi par la fonction différenciée *self-service* qui est également disponible avec WAPT Enterprise.



### 36.1.2 Créer des paquets profile dans la console WAPT

Vous pouvez créer des paquets *profile* en cliquant sur *Dépôt Privé* -> *Générer un modèle de paquet*.



---

**Important :** Pré-requis :

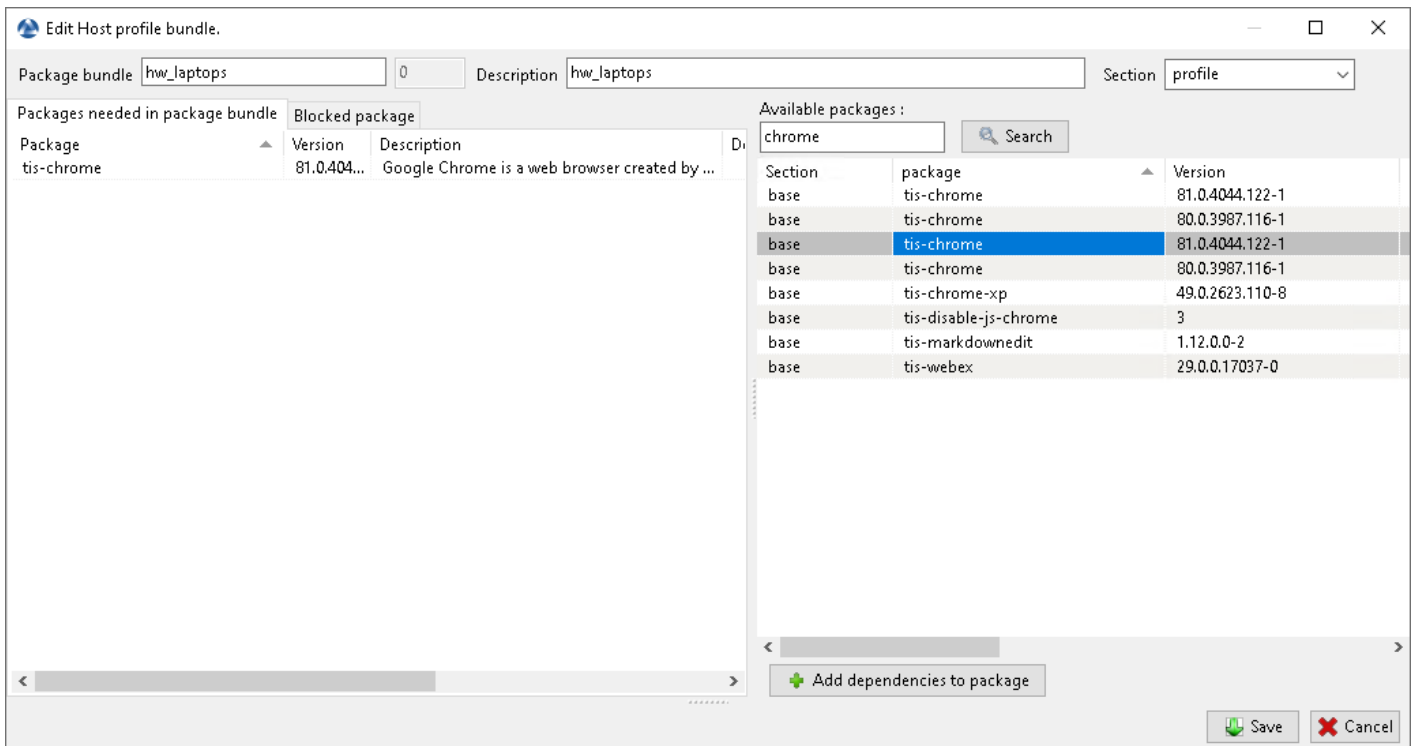
- Le nom du paquet *profile* doit être **exactement** le même que le nom du groupe de sécurité ordinateur de l'AD.
- Le nom du paquet *profile* est **sensible à la casse**.

Exemple :

- Groupe de Sécurité AD : HW\_laptops ;
  - Paquet profile WAPT : HW\_laptops ;
- 

Une fenêtre s'ouvre et on vous demande de choisir quels paquets doivent être contenus dans le paquet **profile** tout juste créé.

Sauvegardez le paquet *profile* qui sera alors téléversé vers le serveur WAPT.



## 36.2 Utiliser des Unités Organisationnelles dans WAPT

### 36.2.1 Principe de fonctionnement

WAPT Enterprise propose la fonctionnalité de paquets d'unité organisationnelle.

Il automatise les installations de logiciels basés sur votre infrastructure Active Directory.

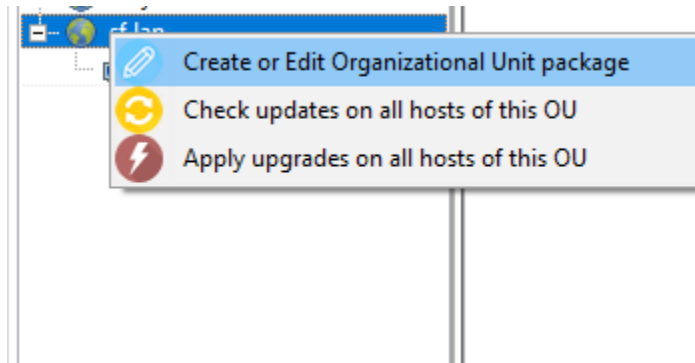
L'agent WAPT connaît son emplacement dans l'arborescence Active Directory, il connaît donc pour cette raison la hiérarchie des Unités Organisationnelles qui le concerne, par exemple :

```
DC=ad,DC=domain,DC=lan
OU=Paris,DC=ad,DC=domain,DC=lan
OU=computers,OU=Paris,DC=ad,DC=domain,DC=lan
OU=service1,OU=computers,OU=Paris,DC=ad,DC=domain,DC=lan
```

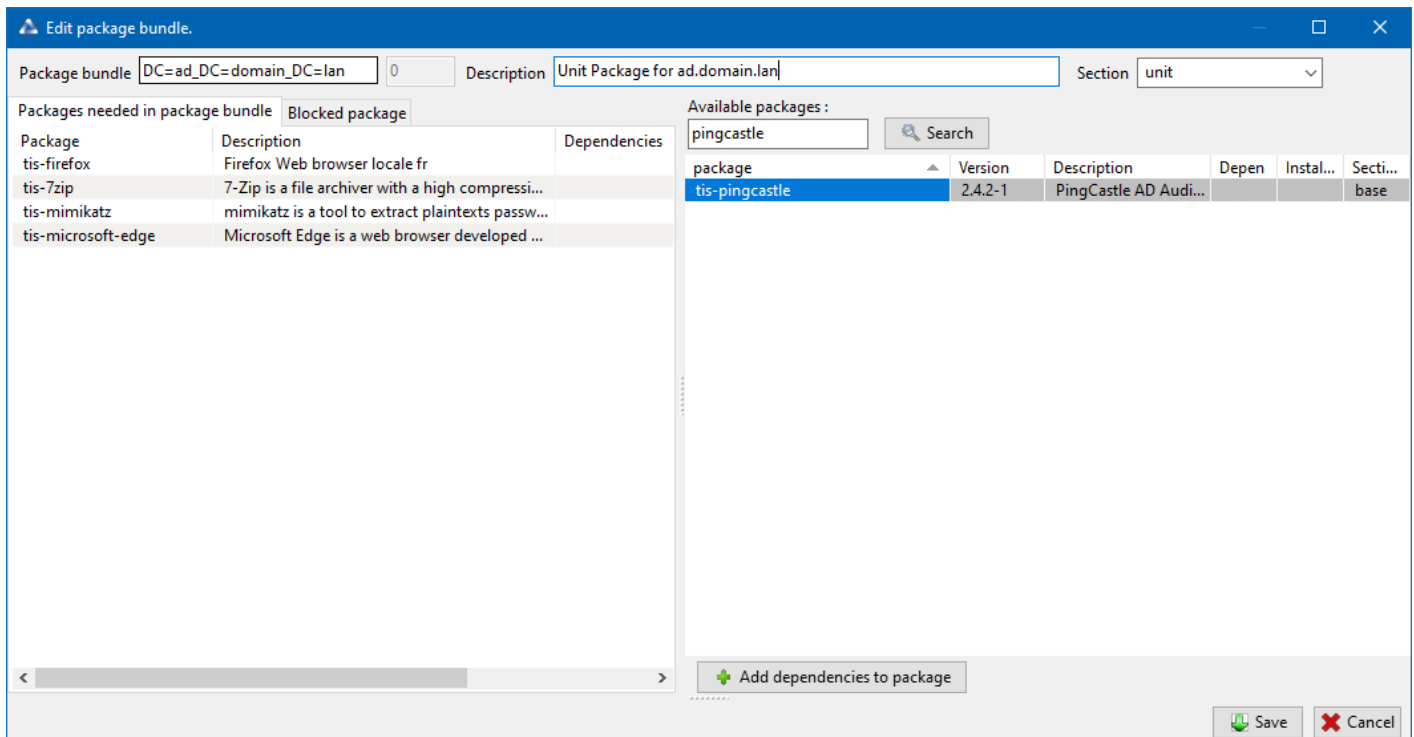
Si un paquet d'Unité Organisationnelle est défini à chaque niveau, l'agent WAPT va automatiquement télécharger des paquets et des configurations qui sont attaché à chaque niveau, par héritage, et va appliquer les paquets attachés ainsi que leurs dépendances.

## 36.2.2 Créer des paquets d'Unité Organisationnelles dans la console WAPT

Vous pouvez créer des paquet *unit* en faisant *clic-droit sur une OU* → *Créer ou éditer le paquet de l'Unité Organisationnelle*.



Une fenêtre s'ouvre et vil vous ai demandé de choisir les paquets qui doivent être inclus dans le paquet **unit**.



Sauvegarder le paquet et il sera déployé sur tous les hôtes appartenant à cette OU.

### 36.2.3 Les actions possibles avec les Unités Organisationnelles

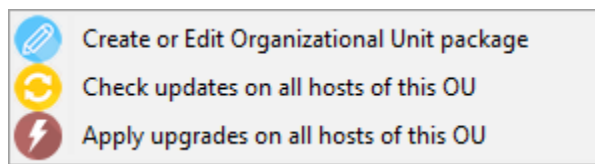
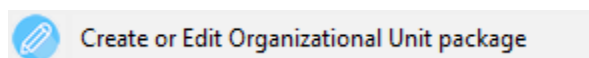


FIG. 1 – La console WAPT affichant les options applicables à l’OU

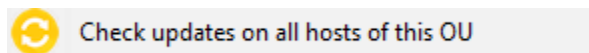
Vous pouvez voir sur l’image que les actions d’*update* et d’*upgrade* peuvent être utilisés à travers ce menu, en sélectionnant ainsi les hôtes par leur unité organisationnelle.

#### Créer ou éditer des paquets d’Unité Organisationnelles



Visitez [cette documentation](#) pour plus de détails sur la Création ou l’Edition de paquet OU.

#### Vérifier les mises à jours des machines de cette OU



Ce bouton va exécuter 2 actions sur tous les hôtes dans l’OU :

1. Donner l’état actuel de l’hôte au serveur.
2. Le serveur affiche si l’hôte doit récupérer des mises à jour.

#### Lancer l’installation des paquets pour les machines de cette OU

Ce bouton permet d’appliquer les mises à jour en attente sur tous les hôtes dans l’OU.

---

**Indication :** Vous pouvez filtrer la manière dont les hôtes sont affichés basé sur leur appartenance a des OU de l’Active Directory.

---

### 36.2.4 Simuler des Unités Organisationnelles pour des hôtes en WORKGROUP

Il arrive que des hôtes spécifiques ne peuvent être joints à un domaine Active Directory.

Avec cette spécificité, de tels hôtes ne peuvent s’afficher dans les Unités Organisationnelles depuis votre console WAPT.

Pour afficher tous les hôtes dans la console sous la bonne Unité Organisationnelle, qu’ils soient joints à un domaine AD ou pas, WAPT vous permet de spécifier une *fausse* Unité Organisationnelle dans le fichier de configuration de l’agent.

Les bénéfices de cette astuce sont :

- Vous pouvez gérer les hôtes avec WAPT comme s’ils étaient joints à l’AD.
- Les hôtes hors-du-domaine et en WORKGROUP s’affiche désormais dans l’arborescence AD.
- Les paquet *Unit* sont utilisables sur ces hôtes.

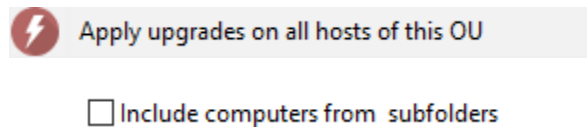


FIG. 2 – La case *Inclure les postes des sous-dossiers* vous permet d’afficher les hôtes des sous-dossiers.

Pour configurer une *fausse* Unité Organisationnelle sur les hôtes, créez un *paquet WAPT vide*, puis utilisez le code suivant :

```
# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = []

def install():

    print('Setting Fake Organizational Unit')
    fake_ou = "OU=TOTO,OU=TEST,DC=DEMO,DC=LAN"
    inifile_writestring(WAPT.config_filename, 'global', 'host_organizational_unit_dn', fake_ou)

    print('Reload WAPT configuration')
    WAPT.reload_config_if_updated()

def update_package():
    pass
```

Le `host_organizational_unit_dn` doit être comme ci-dessus dans `wapt-get.ini` :

```
[global]
host_organizational_unit_dn=OU=TOTO,OU=TEST,DC=DEMO,DC=LAN
```

**Note :**

- Rester sur une casse spécifique avec votre `host_organizational_unit_dn` (ne mélanger pas « dc »s et « DC »s, « ou »s et « OU »s...).
- Continuez à suivre ce chemin dans les champs `DN/computer_ad_dn` dans la grille d’inventaire.

### 36.3 Utiliser les WAPT Windows Update Agent (WAPTWUA)

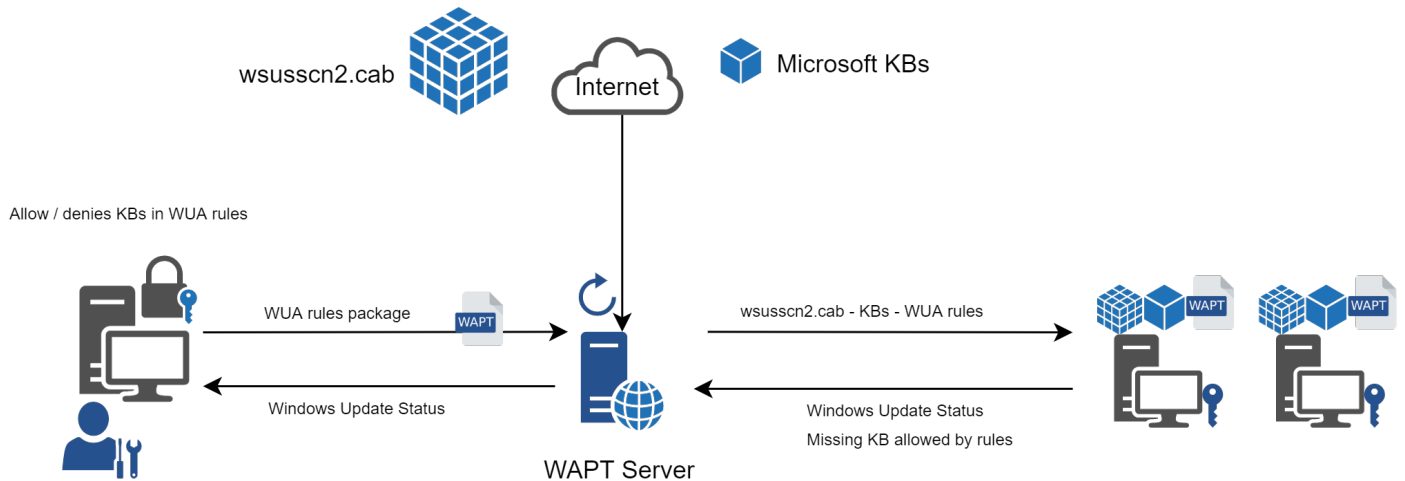
WAPT peut gérer les mises à jours Windows sur vos équipements et remplacer les mises à jour automatiques ou un serveur WSUS.

**Note :** Les composants de WAPTWUA est basé sur l’API WUA (Windows Update Agent). Pour plus d’information : [https://docs.microsoft.com/en-us/windows/win32/wua\\_sdk/using-the-windows-update-agent-api](https://docs.microsoft.com/en-us/windows/win32/wua_sdk/using-the-windows-update-agent-api)

### 36.3.1 Principe de fonctionnement

A chaque PATCH TUESDAY (PATCH TUESDAY EST UN TERME OFFICIEUX POUR SE RÉFÉRER AU DEUXIÈME MARDI DE CHAQUE MOIS QUAND MICROSOFT SORT LES PATCH POUR MEUR LOGICIELS.), LE SERVEUR WAPT TÉLÉCHARGE UN FICHIER :FILE :`WSUSSCN2.CAB à jour depuis les serveurs de Microsoft.

Par défaut, le téléchargement se fait une fois par jour et aucun téléchargement ne se déclenche si le fichier `wsusscn2.cab` n'a pas changé depuis le dernier téléchargement.



Le fichier `wsusscn2.cab` est alors téléchargé par l'agent WAPT depuis le dépôt le plus proche et est ensuite passé au standard de l'utilitaire Windows WUA afin qu'il décortique l'arbre de mise à jour de l'hôte.

Régulièrement, l'hôte va analyser les mises à jour disponibles en se servant du fichier `wsusscn2.cab`. L'hôte enverra sa liste de mises à jour requises déterminées par les WUA vers le serveur WAPT.

Si une mise à jour est en attente sur l'hôte et si cette mise à jour n'est pas présente sur le serveur WAPT, le serveur va télécharger les mises à jours demandées depuis les serveurs officiels Microsoft.

---

**Indication :** Ce mode opératoire permet à WAPT de ne télécharger que les mises à jour nécessaires aux postes, gagnant ainsi de la bande passante, du temps de téléchargement et de l'espace disque.

---

**Note :** Sur le serveur WAPT, les mises à jour téléchargées sont stockées :

- Sous Linux dans `/var/www/waptwua`.
  - Sur les hôtes Windows dans `C:\wapt\waptserver\repository\waptwua`.
- 

Le paramètre de dépôt de l'agent WAPT Windows Update est basé sur l'URL du paramètre `repo_url` dans le `wapt-get.ini` :

- En cas de dépôt répliqué, il est totalement possible de l'utiliser avec WAPT Windows Update afin de réduire l'utilisation de la bande passante.
  - N'oubliez pas de synchroniser le dossier `waptwua` si vous répliquez vos paquets avec des dépôts distants.
- 

**Note :** Si dans votre compagnie un proxy est requis pour sortir sur Internet, alors assurez-vous d'avoir configuré le serveur proxy dans le fichier `waptserver.ini`.

---

### 36.3.2 Les différences entre WAPT Windows Updates et WSUS

WSUS va télécharger par défaut les mises à jours des catégories sélectionnées. Cela peut mener à une base de données vraiment conséquente et une forte utilisation de l'espace de stockage.

WAPT Windows Update ne va que télécharger ce dont il a besoin pour au moins un poste client. Cela aide à garder une base de données locale de petite taille (environ une 10aine de Gigaoctets) et il peut facilement se nettoyer si vous avez besoin de récupérer de l'espace.

### 36.3.3 Les mises à jours majeures d'OS

Les mises à jour majeures d'OS permettent de passer d'une version d'OS à une autre. Cela inclus par exemple, des mises à jours de Windows 7 vers Windows 10, ou bien de Windows 10 1903 vers Windows 10 20H2.

Les mises à jour de version majeures sont gérées de la même manière que les mises à jour d'OS mineures. Les mises à jour majeures sont gérées via le téléchargement du contenu de l'ISO de la nouvelle version (même contenu qu'une installation de base) puis lance le **setup.exe** avec les bons paramètres. Ce Processus est le même que pour WSUS, SCCM et les mises à jour WAPT Windows Updates.

Dans le cas de WAPT Windows Updates, vous aurez besoin de créer un paquet de mise à jour d'OS en utilisant un modèle de paquet fournit sur <https://store.wapt.fr>.

### 36.3.4 Les mises à jours de pilotes

Les mises à jour de pilote via WSUS ne sont pas recommandés puisqu'«il est difficile de gérer correctement les effets de bord. Dans le cas de WAPT WindowsUpdates, **LES PILOTES NE SONT PAS TELECHARGES** puisqu'ils ne sont pas référencés dans les fichiers `wsusscn2.cab` fournis par Microsoft.

Il est recommandé de pousser des mises à jour de pilote via un paquet WAPT personnalisé. Si le patch du pilote est formé en `.msu`, vous pouvez le créer comme un paquet WAPT standard.

Sélectionnez simplement le fichier `.msu` et cliquez sur *Générer un modèle de paquet* depuis la console WAPT pour lancer l'assistant de création de paquet simplifié.

Si la mise à jour du pilote est formé avec un `.zip` contenant le fichier `.exe`, vous pouvez créer un paquet WAPT contenant les fichiers nécessaires et le binaire **setup.exe** avec les options silencieuses correctes.

### 36.3.5 Les KB Out of band (Hors bande)

Parfois, Microsoft fournit des mises à jours OOB (Out of Band) qui sont en dehors de l'index du `wsusscn2.cab`. Ces mises à jour ne sont pas inclus dans les mises à jour principales car elles peuvent corriger un problème très spécifique ou peuvent avoir des inconvénients.

Si vous souhaitez déployer une KB de mise à jour OOB, vous pouvez la télécharger depuis le catalogue Microsoft <https://www.catalog.update.microsoft.com/Home.aspx>.

Sélectionnez simplement le fichier `.msu` et cliquez sur *Générer un modèle de paquet* depuis la console WAPT pour lancer l'assistant de création de paquet simplifié.

Pour ce faire, vous pouvez suivre *cette documentation* pour construire des fichiers `.msu` pour ces mises à jour *Out-of-band (Hors bande)*.

**Attention :** Vous devez vous montrer prudent avec les mises à jour OOB car elles peuvent détruire votre système, assurez-vous de lire les pré-requis sur le rapport Microsoft correspondant à la mise à jour et de tester cette dernière méticuleusement.



### 36.3.6 Configurer WAPTWUA sur l'agent WAPT

WAPTWUA se configure dans `wapt-get.ini` dans la section `[waptwua]`.

Vous aurez alors plusieurs options :

TABLEAU 1 – Les options de configuration dans la section `[waptwua]` dans le `wapt-get.ini`

Options / Valeurs défaut	Description	Exemple
<code>enabled = False</code>	Activer ou désactiver WAPTWUA sur cette machine.	<code>enabled = True</code>
<code>direct_download = False</code>	Télécharger les mises à jour directement depuis les serveurs Microsoft.	<code>direct_download = True</code>
<code>default_allow = False</code>	Configuré si la mise à jour est autorisée ou pas par défaut.	<code>default_allow = True</code>
<code>download_scheduling = None</code>	Configure la récurrence des scans des Windows Update (ne fera rien s'il y a un paquet de règles <code>waptwua</code> ou que le fichier <code>wsusscn2.cab</code> n'a pas changé).	<code>download_scheduling = 1d</code>
<code>install_scheduling = None</code>	Configure la récurrence des installations Windows Update (ne fera rien s'il n'y a aucune mise à jour en attente).	<code>install_scheduling = 2h</code>
<code>install_at_shutdown = False</code>	Installe les mises à jour lorsque la machine s'éteint.	<code>install_at_shutdown = True</code>
<code>install_delay = None</code>	Configure un délai d'installation entre la publication dans le dépôt et l'installation.	<code>install_delay = 15d</code>
<code>allowed_severities = None</code>	Définit une liste de criticité qui sera automatiquement accepté durant un scan WAPT windows update. ex : <i>Important, Critical, Moderate</i> .	<code>allowed_severities = Important</code>

**Indication :** Ces options peuvent être configurées lors de la génération de l'agent.

Exemple de section `[waptwua]` dans le fichier `wapt-get.ini` :

```
[waptwua]
enabled =true
default_allow =False
direct_download=False
download_scheduling=7d
install_at_shutdown=True
install_scheduling=12h
install_delay=3d
```

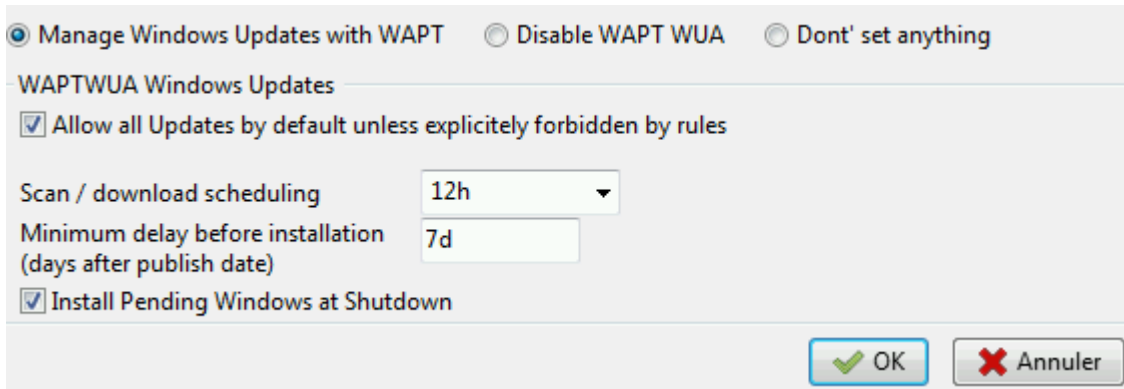
Lorsque vous créez le `waptagent.exe` depuis votre console, ces options correspondent à cela :

Exemple de code source pour un paquet qui modifie les paramètres `[waptwua]` :

```
def install():

    inifile_writestring(WAPT.config_filename, 'waptwua', 'enabled', 'true')
    inifile_writestring(WAPT.config_filename, 'waptwua', 'install_at_shutdown', 'true')
    inifile_writestring(WAPT.config_filename, 'waptwua', 'download_scheduling', '7d')
    inifile_writestring(WAPT.config_filename, 'waptwua', 'allowed_severities', 'Critical,Important')
```

(suite sur la page suivante)



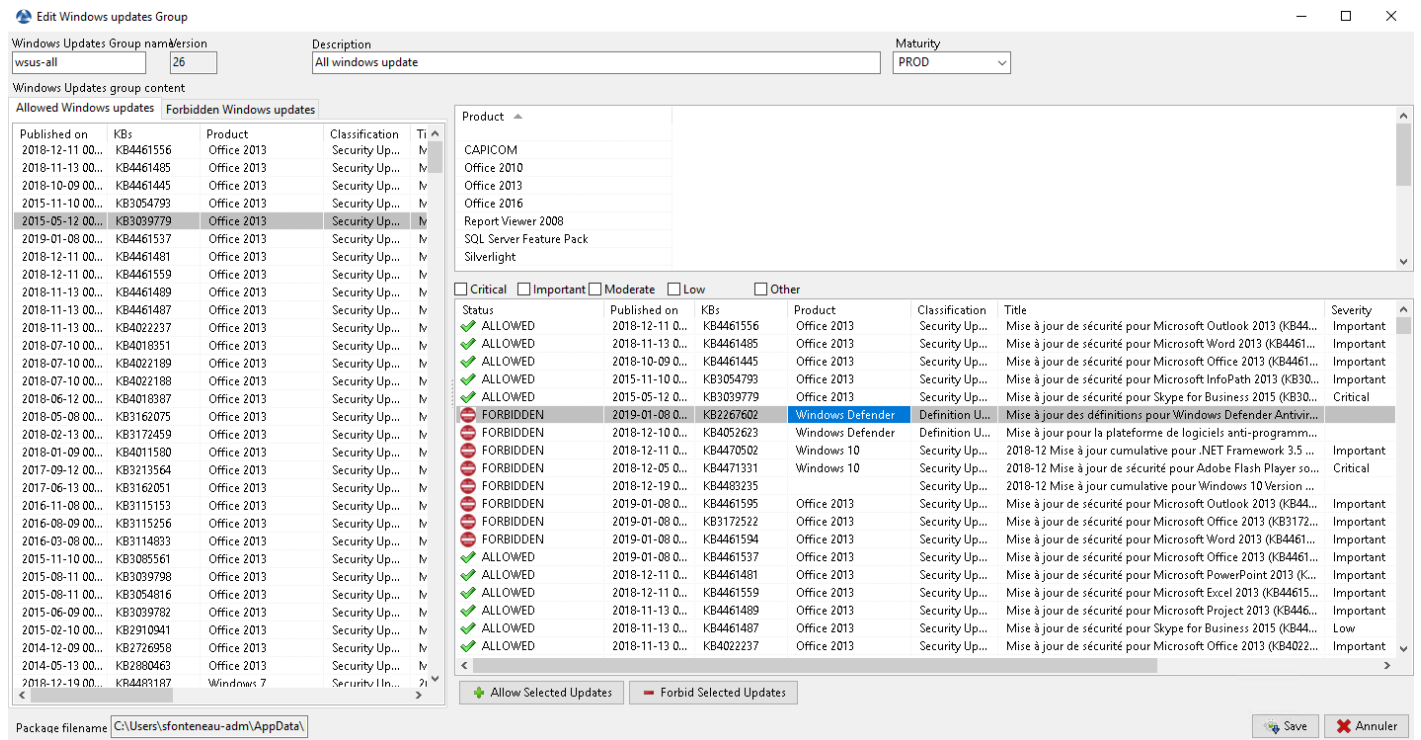
(suite de la page précédente)

```
print('Reload WAPT configuration')
WAPT.reload_config_if_updated()
```

### 36.3.7 Utiliser WAPTWUA depuis la console

Les WAPTWUA sont gérées avec deux onglets dans la console WAPT.

#### Sous-onglet règles WUA dans l'onglet Dépôt privé WAPT



L'onglet *Paquet WAPTWUA* vous permet de créer des paquet de règles *waptwua*.

- Lorsque ce type de paquet est installé sur une machine, il indique à l'agent WAPTWUA les KBs (Knowledge Base articles) autorisées ou interdites.
- Lorsque plusieurs paquets de règles *waptwua* sont installés sur la machine, les différentes règles vont fusionner.
- Lorsqu'une cab n'est ni mentionnée comme autorisée ni mentionnée comme interdite, les agents WAPT vont alors prendre la valeur du `default_allow` dans `wapt-get.ini`.

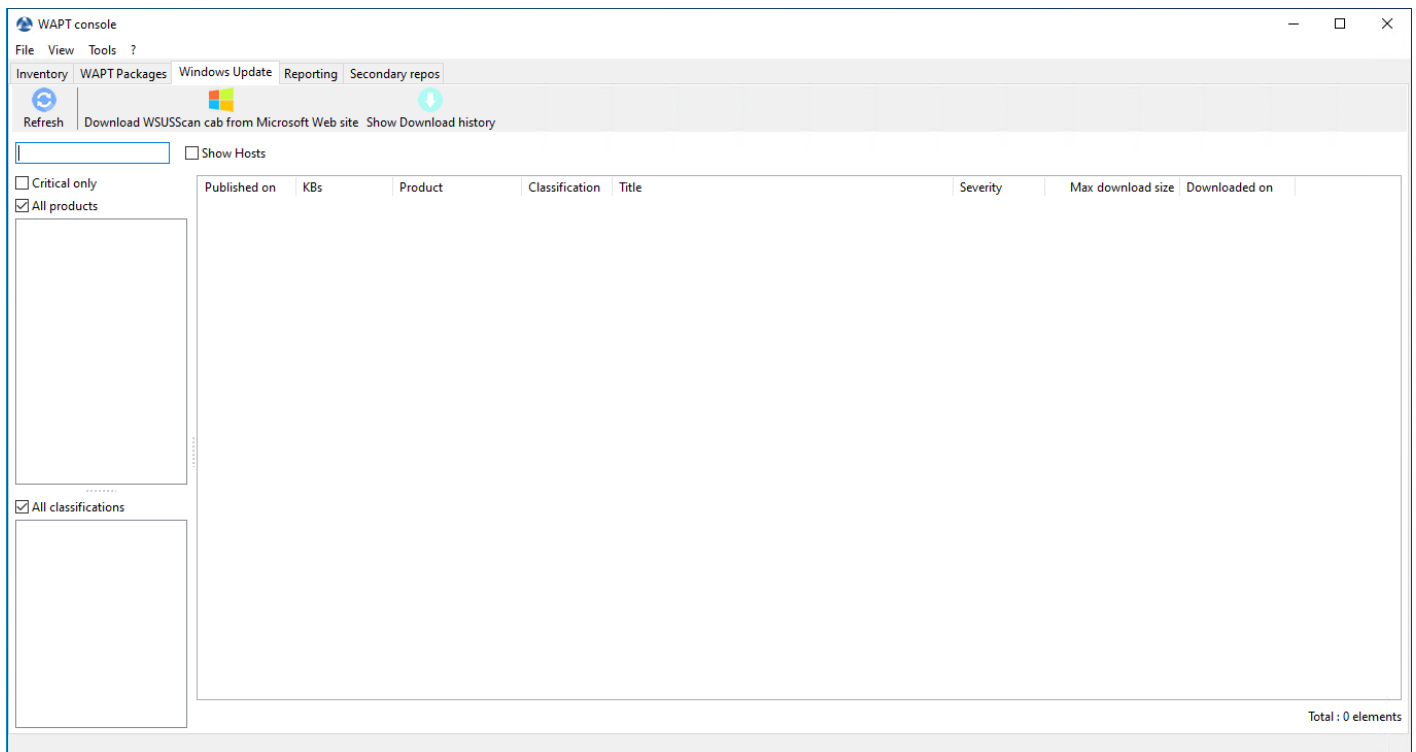
**Note :**

- Si la configuration de l'agent WAPTWUA est défini à `default_allow = True`, alors il sera nécessaire de spécifier les cab interdites.
- Si la configuration de l'agent WAPTWUA est défini à `default_allow = False`, il sera nécessaire de spécifier les cab autorisées.

**Indication :**

- Pour tester les mises à jour sur un petit groupe de postes, vous pouvez configurer la valeur par défaut de WAPTWUA avec `maturity = PREPROD`.
- Vous pouvez tester les mises à jour sur un petit groupe de postes et si tout se passe bien, vous pouvez lancer les mises à jour à votre flotte complète de postes.

## L'onglet liste des Windows Updates



L'onglet *Liste Windows Update* liste toutes les Mises à jour Windows demandées.

**Important :** Le serveur ne scanne pas le fichier `wsussc2.cab` lui-même, il laisse l'Agent Windows Update présent sur toutes les machines Windows le faire. Si une mise à jour vous semble manquante de la liste, vous devez lancer un scan sur une des machines présentes sur la console. Si vous lancez un scan WUA sur un agent Windows 10, le CAB et les fichiers Windows 10 s'afficheront dans l'onglet *Windows Update*.

Le panneau de gauche affiche les catégories des mises à jour, vous permettant de filtrer par :

- Criticité ;
- Produit ;
- Classification ;

Dans le panneau de droite, si la colonne *Téléchargée* est vide, cela signifie que les mises à jour n'ont pas encore été téléchargées par le serveur WAPT et n'est pas présent sur le serveur WAPT (Cette mise à jour n'est pas manquante sur les postes).

- Vous pouvez forcer le téléchargement de la mise à jour en faisant *clic-droit* → *Télécharger les mises à jour sélectionnées*.
- Vous pouvez aussi forcer le téléchargement du fichier `wsussc2.cab` avec le bouton *Télécharger le cab WSUSScan depuis le site de Microsoft*.
- Vous pouvez voir le téléchargement des mises à jour Windows sur le serveur avec le bouton *Afficher la tâche de téléchargement*

Published on	KBs	Product	Classification	Title	Severity	Max download size	Downloaded on
2017-06-27	KB2656356	Windows 7	Mise à jour ...	Mise à jour de sécurité pour Microsoft .NET Framework 3.5...	Critical	4.1 MB	
2021-09-14	KB5005565	Windows 10, versio...	Security Up...	2021-09 Mise à jour cumulative pour Windows 10 Version ...	Critical	586.8 MB	
2021-09-14	KB5005565	Windows 10, versio...	Security Up...	2021-09 Mise à jour cumulative pour Windows 10 Version ...	Critical	586.8 MB	
2021-09-13	KB5005565	Windows 10, versio...	Mise à jour ...	2021-09 Mise à jour cumulative pour Windows 10 Version ...	Critical	586.8 MB	
2021-03-08	KB5000808	Windows 10, versio...	Security Up...	2021-03 Mise à jour cumulative pour Windows 10 Version ...	Critical	458.4 MB	2021-09-09 17:31...
2020-08-10	KB4569751	Windows 10, versio...	Security Up...	2020-08 Mise à jour cumulative pour .NET Framework 3.5 p...	Critical	75.6 MB	2021-09-09 17:30...
2020-07-10	KB4565633	Windows 10, versio...	Security Up...	2020-07 Mise à jour cumulative pour .NET Framework 3.5 p...	Critical	71.3 MB	
2020-01-09	KB4532938	Windows 10, versio...	Security Up...	2020-01 Mise à jour cumulative pour .NET Framework 3.5 p...	Critical	67.8 MB	
2021-08-09	KB5005260	Windows 10, versio...	Mise à jour ...	2021-08 Mise à jour de la pile de maintenance pour Windo...	Critical	14.6 MB	
2021-08-09	KB5005033	Windows 10, versio...	Mise à jour ...	2021-08 Mise à jour cumulative pour Windows 10 Version ...	Critical	593.6 MB	
2021-08-09	KB5005260	Windows 10, versio...	Security Up...	2021-08 Mise à jour de la pile de maintenance pour Windo...	Critical	14.6 MB	
2021-08-09	KB5005033	Windows 10, versio...	Security Up...	2021-08 Mise à jour cumulative pour Windows 10 Version ...	Critical	593.6 MB	
2021-08-09	KB5005260	Windows 10, versio...	Security Up...	2021-08 Mise à jour de la pile de maintenance pour Windo...	Critical	14.6 MB	
2021-08-09	KB5005033	Windows 10, versio...	Security Up...	2021-08 Mise à jour cumulative pour Windows 10 Version ...	Critical	593.6 MB	
2021-07-12	KB5004748	Windows 10, versio...	Security Up...	2021-07 Mise à jour de la pile de maintenance pour Windo...	Critical	14.4 MB	
2021-07-12	KB5004245	Windows 10, versio...	Security Up...	2021-07 Mise à jour cumulative pour Windows 10 Version ...	Critical	525.4 MB	
2021-07-12	KB5004237	Windows 10, versio...	Security Up...	2021-07 Mise à jour cumulative pour Windows 10 Version ...	Critical	584.9 MB	
2021-07-12	KB5004237	Windows 10, versio...	Security Up...	2021-07 Mise à jour cumulative pour Windows 10 Version ...	Critical	584.9 MB	2021-08-11 10:27...
2021-07-12	KB5004237	Windows 10, versio...	Security Up...	2021-07 Cumulative Update for Windows 10 Version 2004 f...	Critical	584.9 MB	2021-08-11 10:27...
2021-07-06	KB5004945	Windows 10, versio...	Security Up...	2021-07 Mise à jour cumulative pour Windows 10 Version ...	Critical	583.3 MB	
2021-06-08	KB5003711	Windows Server 2019	Security Up...	2021-06 Mise à jour de la pile de maintenance pour Windo...	Critical	13.6 MB	2021-08-03 12:20...
2021-06-08	KB5003671	Windows Server 20...	Security Up...	2021-06 Correctif cumulatif mensuel de qualité pour Wind...	Critical	532.7 MB	2021-08-03 12:20...
2021-06-08	KB5003681	Windows Server 20...	Security Up...	2021-06 Mise à jour qualitative de sécurité uniquement po...	Critical	36.6 MB	2021-08-03 12:17...
2021-05-10	KB5003220	Windows Server 20...	Security Up...	2021-05 Mise à jour qualitative de sécurité uniquement po...	Critical	27.6 MB	2021-07-02 11:47...
2021-05-10	KB5003169	Windows 10, versio...	Security Up...	2021-05 Mise à jour cumulative pour Windows 10 Version ...	Critical	563.8 MB	
2021-05-10	KB5003244	Windows 10, versio...	Security Up...	2021-05 Mise à jour de la pile de maintenance pour Windo...	Critical	14.3 MB	
2021-04-12	KB5001393	Windows Server 20...	Security Up...	2021-04 Mise à jour qualitative de sécurité uniquement po...	Critical	53.5 MB	2021-07-02 11:47...

Total : 123 elements

**Indication :** Toutes les 30 minutes, le serveur WAPT va chercher les mises à jour qui ont été demandées au moins une fois par le client WAPT et qui n'ont pas été téléchargées en mises en cache. » Si une mise à jour est en attente, le serveur WAPT va le télécharger depuis les sites officiels de Microsoft.

Vous pouvez forcer ce scan avec le bouton *Télécharger le cab WSUSScan depuis le site de Microsoft*; dans l'onglet *Mises à jour Windows* → *Liste Windows Updates*

## Nettoyer des vieilles Windows Update

### Automatiquement

Si la KB n'est pas installée sur l'hôte, elle est automatiquement supprimée sur le Server WAPT entre 2h30 et 3h30 tous les jours.

Il est possible de désactiver l'autowash sur `waptserver.ini` avec l'option `cleanup_kbs`.

Ajouter le paramètre dans le fichier de configuration du serveur :

```
cleanup_kbs = False
```

### Sur la console

Pour nettoyer votre dossier `waptwua`, vous pouvez aller dans l'onglet *Mises à jour Windows* et cliquer sur *Effacer les mises à jour non-affectées*. Cela supprimera toutes les kb inutilisées stockées dans votre serveur WAPT.



### Sur le serveur

L'ancienne méthode pour ce faire, est de supprimer manuellement les mises à jour Windows qui ne sont plus utilisées directement sur le serveur.

Le serveur WAPT va seulement re-télécharger les mises à jour supprimées si un des hôtes équipés le demande.

Sur le serveur WAPT, les mises à jour téléchargées sont stockées :

- Sous Linux dans `/var/www/waptwua`.
- Sur les hôtes Windows dans `C:\wapt\waptserver\repository\waptwua`.

### Lancer WUA sur les clients

Depuis la console vous avez trois options.

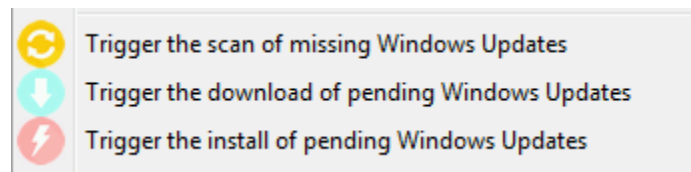


FIG. 3 – Les boutons d'action Windows Update disponibles depuis la console WAPT

- Le bouton *Lancer la recherche de mises à jour* va lancer le scan sur le client et va lister toutes les mises à jour marquées pour l'OS.
- Le bouton *Lancer le téléchargement des Mises à jour Windows en attente* va lancer le téléchargement des mises à jour en attente sur le client.
- Le bouton *Lancer l'installation des mises à jour Windows en attente* va lancer l'installation des mises à jour téléchargées sur le client.

**Indication :** Lorsque des mises à jour en attente sont stockées en cache pour être installées, l'agent WAPT va déclencher le service WUA.

L'agent WAPT va activer et démarrer le service WUA temporairement pour installer les mises à jour. Lorsque les mises à jour sont installées, wapt-service va couper et désactiver le service WUA jusqu'au prochain cycle.

### Etat des mises à jour sur l'hôte

Les mises à jour Windows peuvent avoir 4 états sur un poste.

Statut	Description
<i>OK</i>	Une mise à jour Windows qui s'est correctement installée.
<i>MISSING</i>	Une mise à jour Windows qui n'a pas encore été téléchargée sur le serveur WAPT.
<i>PENDING</i>	Le serveur WAPT sait qu'il doit télécharger une mise à jour depuis les serveurs officiels de Microsoft.
<i>DISCARDED</i>	Une mise à jour Windows interdites par des règles.

FIG. 4 – Les mises à jour Windows en attente affichées dans la console WAPT

### Notion d'UpdateID

Dans WAPT nous n'utilisons pas les kbids mais les **updateids**.

Cela nous permet d'être plus fin dans la gestion des mises à jour.

ID Mise à jour	Publiée le	KBs	Produit	Classification	Titre
0fc3c864-ee8f-4166-8889-2d2bfc70000e_200	2020-02-10	KB4537759	Windows 10	Security Updates	2020-02 Mise à jour de sécurité pour Adobe Flash Player sous Windows 10 Version 1803 sur systèmes x64 (KB4537759)
ad555e0c-f639-463a-b4ec-04e9209aff2_200	2020-02-10	KB4537759	Windows 10, version 1903 and later	Security Updates	2020-02 Mise à jour de sécurité pour Adobe Flash Player sous Windows 10 Version 1909 sur systèmes x64 (KB4537759)
3e6c0dae-aa30-4f85-ba1e-9b698eb2c374_200	2020-02-10	KB4537759	Windows 10, version 1903 and later	Security Updates	2020-02 Mise à jour de sécurité pour Adobe Flash Player sous Windows 10 Version 1903 sur systèmes x64 (KB4537759)

FIG. 5 – Dupliquer la kb

Dans cet exemple, la KB4537759 apparaît de multiples fois car il y a 3 différents updateids :

- win10 1803 ;
- win10 1903 ;
- win10 1909 ;

Vous devriez également autoriser une mise à jour et non pas une *kb ids*.

### 36.3.8 WAPT ne force pas Windows à désinstaller une Windows Update

Désinstaller une mise à jour Windows peut être dangereux pour la machine. Quand une mise à jour est détectée comme interdite par WAPT, sa désinstallation ne sera **PAS** forcée.

Si vous voulez vraiment désinstaller une mise à jour, vous devriez créer un paquet pour désinstaller la KB.

Voici un exemple :

```
from setuphelpers import *

uninstallkey = []

def install():
    with EnsureWUAServRunning():
        run('wusa /uninstall /KB:4023057')
```

### 36.3.9 Vidéo de démonstration

## 36.4 Utiliser des paquets profile dans WAPT

### 36.4.1 Principe de fonctionnement

WAPT **Enterprise** offre des fonctionnalités de reporting avancées.

En effet, qui mieux que vous pouvez savoir ce dont vous avez besoin dans votre rapport.

Avec WAPT nous vous proposons d'écrire vos requêtes SQL dont le résultat s'affichera dans la console WAPT.

### 36.4.2 Concepteur de requêtes WAPT

Le concepteur de requêtes vous offre la possibilité de modifier vos propres requêtes sur la base de données PostgreSQL de WAPT.

Pour créer une nouvelle requête, cliquez sur *Reporting* → *Mode conception* → *Nouvelle requête*.

---

#### Indication :

- Pour renommer une requête, appuyez sur la touche F2.
  - Dans l'encadré du haut, vous pouvez écrire votre requête SQL.
- 

Pour éditer / modifier / Sauvegarder vos requêtes :

- Le bouton *Recharger* est utilisé pour recharger les requêtes sur le serveur, par exemple, si un collègue vient juste d'éditer une nouvelle requête.
  - Le bouton *Nouvelle requête* va ajouter une requête vide à la liste.
  - Le bouton *Supprimer la requête* va supprimer la requête sélectionnée sur le serveur WAPT.
  - Le bouton *Exporter vers tableur* va exporter le résultat de votre requête dans un feuille de calcul.
  - Le bouton *Enregistrer tout* va sauvegarder votre requête au serveur WAPT.
  - Le bouton *Dupliquer* va dupliquer une requête existante pour éviter de repartir d'une requête vide.
  - Le bouton *Exécuter* va exécuter la requête sélectionnée.
- 

**Note :**

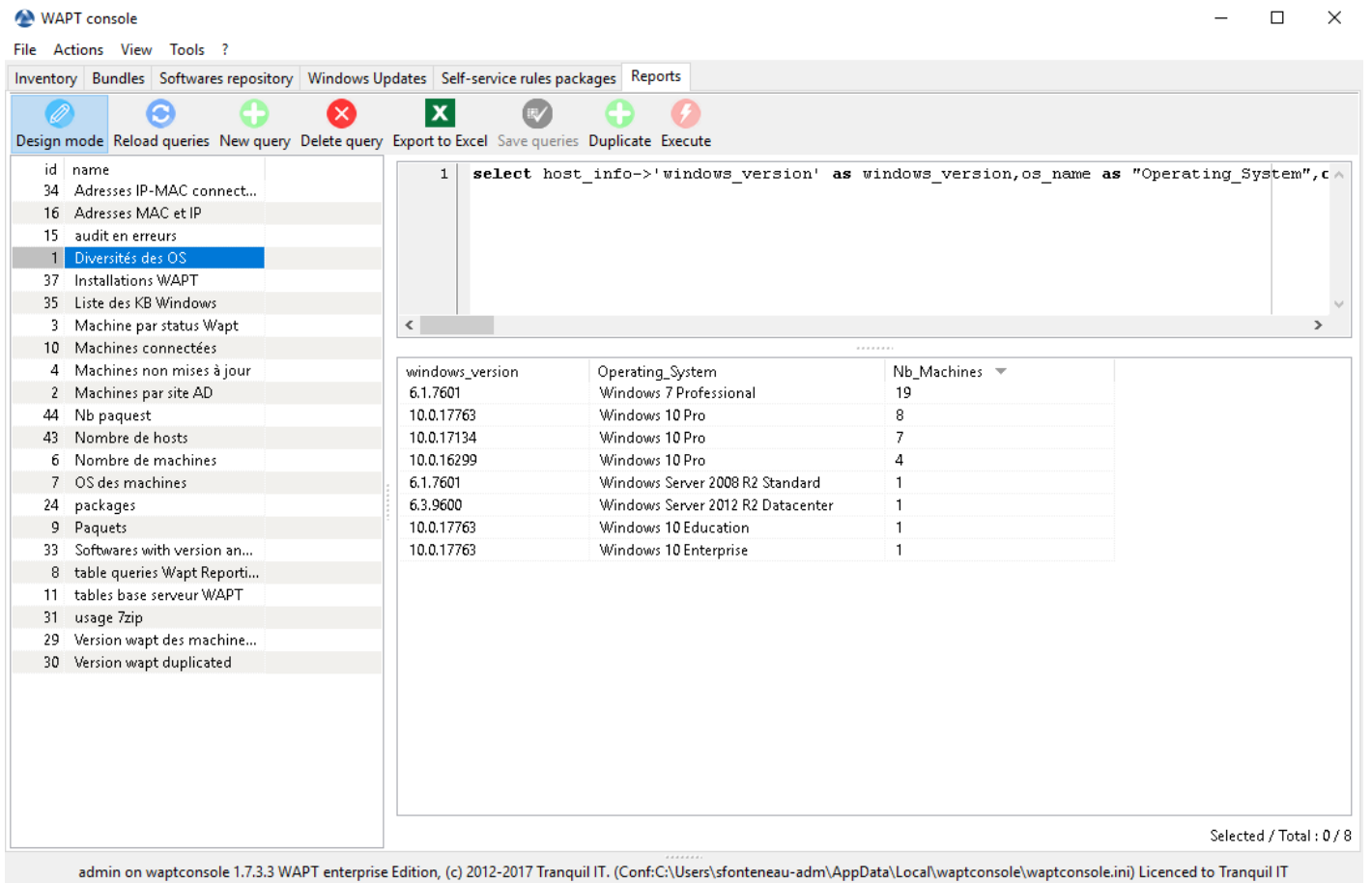


FIG. 6 – Concevoir une requête dans le reporting WAPT



- Les requêtes sont sauvegardées dans la base de données PostgreSQL WAPT.
- Le raccourci CTRL+espace vous permet de construire votre requête de façon plus efficace.

### 36.4.3 Exemple de requêtes

#### Requêtes Ordinateur

- Compter les hôtes.

```
select count(*) as "Nb_Machines" from hosts
```

- Lister les postes.

```
select
computer_name,
os_name,
os_version,
os_architecture,
serialnr
from hosts
order by 4,3,1
```

- Lister les adresses MAC et Ip des postes.

```
select distinct unnest(mac_addresses) as mac,
unnest(h.connected_ips) as ipaddress,
computer_fqdn,h.description,
h.manufacturer||' '||h.productname as model,
h.serialnr,
h.computer_type
from hosts h
order by 1,2,3
```

- Lister la version de Windows.

```
select
host_info->'windows_version' as windows_version,
os_name as operating_system,
count(os_name) as nb_hosts
from hosts
group by 1,2
```

- Lister les systèmes d'exploitation.

```
select host_info->'windows_version' as windows_version,
os_name as "Operating_System",
count(os_name) as "Nb_Machines"
from hosts
group by 1,2
```

- Lister les postes qui ne sont pas vus depuis un moment.

```
select
h.uuid,
h.computer_fqdn,
install_date::date,
version,
h.listening_timestamp::timestamp,
h.connected_users from hostsoftwares s
left join hosts h on h.uuid=s.host_id
where s.key='WAPT_is1'
and h.listening_timestamp<'20190115'
```

— Filtrer les postes pas types de chassis.

```
select case
dmi->'Chassis_Information'->>'Type'
when 'Portable' then '01-Laptop'
when 'Notebook' then '01-Laptop'
when 'Laptop' then '01-Laptop'
when 'Desktop' then '02-Desktop'
when 'Tower' then '02-Desktop'
when 'Mini Tower' then '02-Desktop'
else '99-' || (dmi->'Chassis_Information'->>'Type')
end as type_chassis,
string_agg(distinct coalesce(manufacturer, '?') || ' ' || coalesce(productname, ''), ', ' ),
count(*) as "Nb_Machines" from hosts
group by 1
```

— Lister les postes avec leur Numéro de Série Windows.

```
select
computer_name,
os_name,
os_version,
host_info->'windows_product_infos'->'product_key' as windows_product_key
from hosts
order by 3,1
```

### Requête WAPT

— Lister les paquets WAPT du dépôt du serveur WAPT.

```
select
package,
version,
architecture,
description,
section,
package_uuid,
count(*)
```

(suite sur la page suivante)

(suite de la page précédente)

```
from packages
group by 1,2,3,4,5,6
```

— Lister les postes nécessitant une mise à jour.

```
select
computer_fqdn,
host_status,
last_seen_on::date,
h.wapt_status,
string_agg(distinct lower(s.package), ' ')
from hosts h
left join hostpackagesstatus s on s.host_id=h.uuid and s.install_status != 'OK'
where (last_seen_on::date > (current_timestamp - interval '1 week')::date
and host_status!='OK')
group by 1,2,3,4
```

### Requête Paquets

— Lister les paquets avec leur nombre d'installation.

```
select
package,
version,
architecture,
description,
section,
package_uuid,
count(*)
from hostpackagesstatus s
where section not in ('host', 'unit', 'group')
group by 1,2,3,4,5,6
```

### Requête logiciel

— Lister les agents WAPT Discovery.

```
select
h.uuid,
h.computer_name,
install_date::date,
version,
h.listening_timestamp::timestamp,
name
from hostsoftwares s
left join hosts h on h.uuid=s.host_id
where
```

(suite sur la page suivante)

```
s.key='WAPT_is1'
and (name ilike 'WAPT%Discovery%' or name ilike 'WAPT %')
```

— Lister les postes avec leur version de 7zip associé.

```
select
hosts.computer_name,
hostsoftwares.host_id,
hostsoftwares.name,
hostsoftwares.version
from hosts, hostsoftwares
where hostsoftwares.name ilike '7-zip%'
and hosts.uuid=hostsoftwares.host_id
order by hosts.computer_name asc
```

— Lister les postes avec leurs logiciels.

```
select
n.normalized_name,
s.version, string_agg(distinct lower(h.computer_name), ' '),
count(distinct h.uuid)
from hostsoftwares s
left join normalization n on (n.original_name = s.name) and (n.key = s.key)
left join hosts h on h.uuid = s.host_id
where (n.normalized_name is not null)
and (n.normalized_name <> '')
and not n.windows_update
and not n.banned
and (last_seen_on::date > (current_timestamp - interval '3 week')::date)
group by 1,2
```

— Lister les logiciels normalisés.

```
select
n.normalized_name,
string_agg(distinct lower(h.computer_name), ' '),
count(distinct h.uuid)
from hostsoftwares s
left join normalization n on (n.original_name = s.name) and (n.key = s.key)
left join hosts h on h.uuid = s.host_id
where (n.normalized_name is not null)
and (n.normalized_name <> '')
and not n.windows_update
and not n.banned
and (last_seen_on::date > (current_timestamp - interval '3 week')::date)
group by 1
```

Vous pouvez aussi trouver plus d'exemple de requêtes sur le [Forum Tranquil IT](#).

N'hésitez pas à partager vos requêtes sur le même forum avec une explication de ce que fait votre requête, idéalement avec une capture d'écran ou une table affichant un échantillon du résultat de votre requête.

### 36.4.4 Normaliser les noms de logiciels

Parfois, la version du logiciel ou son architecture fait partie intégrante du nom du logiciel. Quand le logiciel s'enregistre dans l'inventaire du serveur WAPT, il apparaît en différents logiciels alors qu'ils sont pareils pour nous humains.

Pour résoudre ce problème, nous proposons de standardiser le nom des logiciels avec WAPT.

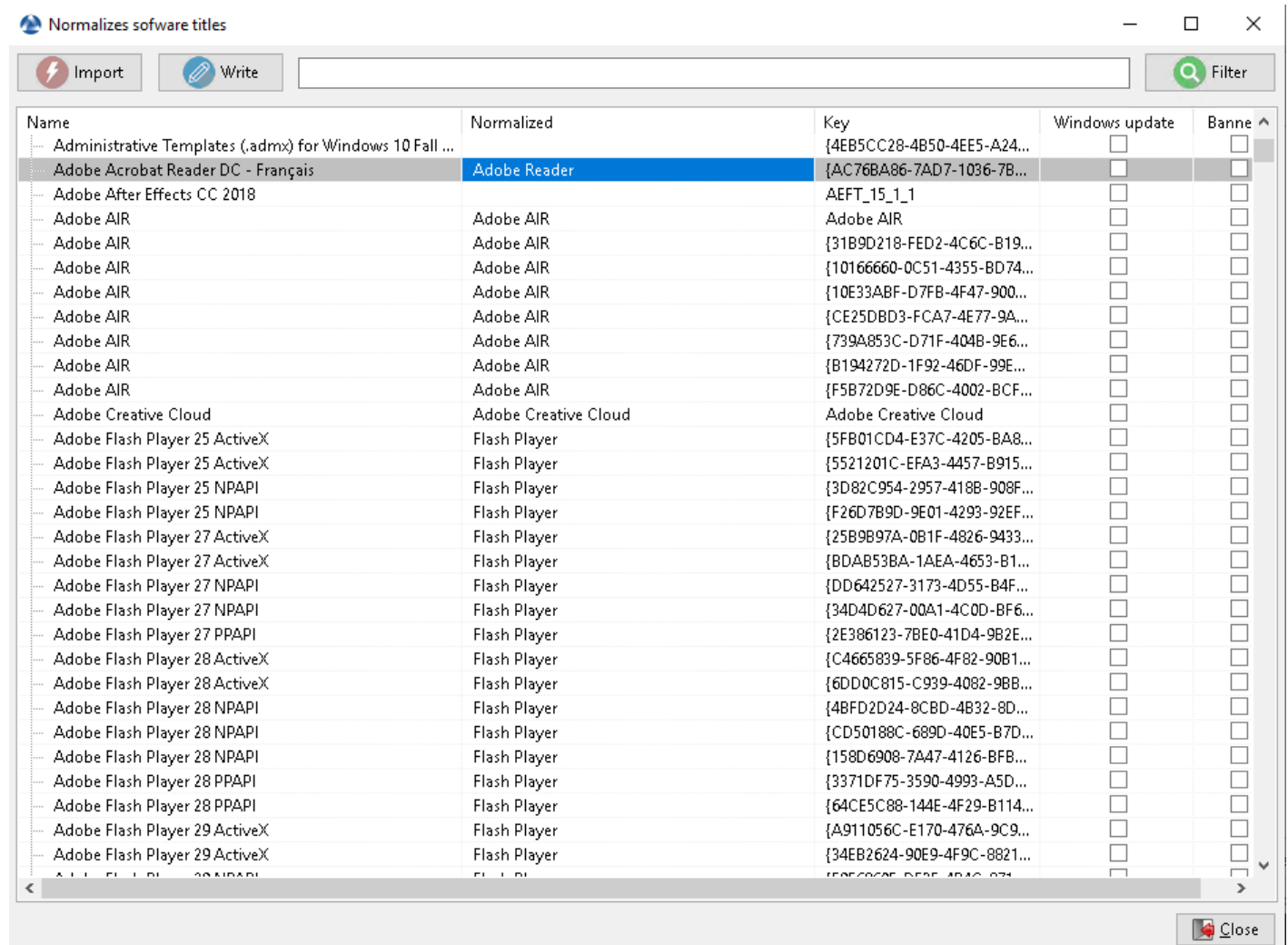


Fig. 7 – Normaliser le nom du logiciel

- Cliquez sur *Normaliser les nom de logiciels* dans le menu *Outils*.
- Sélectionnez le logiciel à standardiser, par exemple, toutes les version différentes d'Adobe Flash player.
- Sur la colonne *Normalisé*, appuyez sur F2 pour assigner un nom standard sur le logiciel sélectionné. Puis appuyez sur Entrée.

#### Note :

- Pour sélectionner plusieurs programmes, sélectionnez les avec les combinaisons de touches *shift-up/down*.
- Vous pouvez aussi marquer un logiciel comme *Mise à jour Windows* ou *Banni* (Appuyez sur la barre espace dans la colonne correspondante).

- Appuyez sur *Importer* pour charger les changements du serveur.
- Appuyez sur *Enregistrer* pour sauvegarder vos changements.

Vous pouvez maintenant lancer vos requêtes avec ce nom standardisé.

## 36.5 Se connecter à la base de données WAPT avec un client PostgreSQL

Vous pouvez connecter votre base de données WAPT à un client si vous préférez utiliser un client PostgreSQL.

Pour ce faire, vous allez devoir changer quelques fichiers de configuration sur votre serveur WAPT.

- Trouvez dans quelle version est votre PostgreSQL.

```
ps -ef | grep -i sql
postgres  512      1  0 Jan05 ?          00:00:24 /usr/lib/postgresql/12/bin/postgres -D /var/
↳ lib/postgresql/12/main -c config_file=/etc/postgresql/12/main/postgresql.conf
```

- Modifiez `pg_hba.conf` de la version PostgreSQL utilisée. Dans `/etc/postgresql/12/main/pg_hba.conf` pour Debian et `/var/lib/pgsql/12/data/pg_hba.conf` pour Centos sous # **IPv4 local connections section**, ajoutez votre adresse.

```
host      wapt          all          192.168.0.65/32          md5
```

where 192.168.0.65 is your IP address that is authorized to connect to the WAPT database.

- Autorisez PostgreSQL à écouter sur toutes les interfaces dans `/etc/postgresql/12/main/postgresql.conf` pour Debian et `/var/lib/pgsql/12/data/postgresql.conf` pour Centos, section **Connection Settings**.

```
listen_addresses = '*'
```

- Redémarrez le service pour votre version de PostgreSQL.

```
systemctl restart postgresql@12-main.service
```

- Pour se connecter au PostgreSQL sur le serveur wapt.

```
sudo -u postgres psql template1
```

- Puis renseignez le mot de passe de l'utilisateur wapt.

```
template1=# ALTER USER wapt WITH PASSWORD 'PASSWORD';
```

### 36.5.1 Vidéo de démonstration

## 36.6 Synchroniser les inventaires de WAPT vers GLPI

### 36.6.1 Principe de fonctionnement

WAPT Enterprise propose une synchronisation entre les inventaires de vos postes et GLPI ITSM Software.

Cette méthode synchronise automatiquement les changements sur votre infrastructure informatique avec le serveur GLPI.

**Attention :** GLPI par WAPT ne fonctionne pas avec l'authentification Kerberos pour GLPI.

Si vous utilisez Kerberos pour GLPI, excluez `glpi/plugins/fusioninventory/` de l'authentification NGINX.

### 36.6.2 Installer les dépendances requises

Pour pouvoir recevoir les inventaires sur votre serveur GLPI, vous aurez besoin du plugin **FusionInventory** sur votre serveur GLPI.

**Note :** Vous pouvez suivre ce guide pour installer FusionInventory.

Après avoir installé FusionInventory, vous aurez un **point d'accès** sur votre serveur WAPT pour envoyer les inventaires vers :

`http://glpi.mydomain.lan/glpi/plugins/fusioninventory/`

### 36.6.3 Configuration

Vous pouvez ouvrir une fenêtre pour configurer GLPI avec *Outils > Exporter les données d'inventaire vers GLPI*.

In *Glpi Server Properties*, add the required parameters in the `waptserver.ini` configuration file.

```
[options]
...
glpi_server_endpoint = http://glpi.mydomain.lan/glpi/plugins/fusioninventory/
glpi_server_user = user
glpi_server_pass = password
glpi_server_pause_timeout = 20,15
glpi_inventory_update_delay = 4
glpi_inventory_update_range = 25
```

- `glpi_server_endpoint` : l'url vers le plugin FusionInventory où l'inventaire sera téléversé sur le serveur GLPI ;
- `glpi_server_user`, `glpi_server_pass` : Les accès de GLPI ;
- `glpi_server_pause_timeout = A,B` : pause le téléversement pendant **A** secondes lorsque le serveur prend plus **B** secondes à répondre ;
- `glpi_inventory_update_range` : à combien de téléversement vous voulez que la base de données soit mise à jour, si vous arrêtez le téléversement, cela va redémarrer au à la dernière mise à jour ;
- `glpi_inventory_update_delay = C` : Le téléversement est déclenché automatiquement toutes les **C** heures si cela n'est pas déjà en cours ;
- `glpi_inventory_debug_directory = exemple /tmp/glpi` répertoire où tous les fichiers d'inventaire XML seront stockés et téléversés sur le serveur GLPI (le nom du fichier est le GUID de la machine avec l'extension `.xml`). Il est possible d'en utiliser les contenus pour diagnostiquer les problèmes tels que les divergences entre les valeurs attendues dans GLPI et celles vues actuellement dans WAPT.

### 36.6.4 Utiliser WAPT pour envoyer les mises à jour d'inventaire vers GLPI

Comme vous pouvez le voir sur la *fenêtre de configuration*, vous pouvez remplir les paramètres, déclencher ou arrêter un téléversement directement depuis la console WAPT :

- Lorsque vous remplissez les *Propriétés du serveur GLPI*, la configuration est enregistrée sur le serveur WAPT lorsque vous cliquez sur *Sauvegarder*.
- Vous pouvez récupérer des propriétés déjà enregistrées sur le serveur WAPT en cliquant sur *Reset* (le mot de passe n'est pas chargé).
- Le champ *Point de connexion* est l'url du serveur GLPI servant à envoyer les inventaires vers le `glpi_server_endpoint`.
- Vous pouvez déclencher un téléversement sans attendre la tâche planifiée en cliquant sur *Envoi des données*.
- Vous pouvez arrêter le téléversement n'importe quand en cliquant sur *Annuler*.

---

**Indication :** Le status du téléversement est mis à jour toutes les 15s, vous pouvez suivre l'état avec la barre de progression.

---

Si vous avez beaucoup de postes, le téléversement peut prendre longtemps. Afin d'éviter cela, lorsque le téléversement est lancé, seuls les inventaires qui ont changé sont téléversés :

- Avec `Téléchargement forcé`, chaque inventaire est téléversé, ignorant les données déjà téléversées.

### 36.6.5 Utilisation avancée du plugin GLPI

Pour afficher les propriétés avancées, sélectionnez le bouton *Avancé*.

The screenshot shows a configuration window with a checked checkbox labeled "Avancé". Below it is a section titled "Propriétés avancées" containing three input fields with labels and units:

- "Tâche automatique toutes les:" followed by a text input containing "4" and the unit "heures".
- "Pause de:" followed by a text input containing "20" and "s au-delà de:" followed by a text input containing "15" and "s de délai d'expiration".
- "MàJ de la bdd tout les:" followed by a text input containing "25" and "téléchargements".

At the bottom right of the configuration area are two buttons: "Réinitialiser" and "Enregistrer".

- La tâche planifiée tourne toutes les *Cron every...* heures seulement si le *Point de connexion* est renseigné. Vous pouvez désactiver la tâche planifiée en laissant le *Point de connexion* vide. ~> `glpi_inventory_update_delay`.



**Indication :** Si vous voulez désactiver le téléversement automatique, vous devez *Sauvegarder* un *Point de connexion* vide.

- Vous pouvez déclencher des pauses (*Pause...*) lorsque le temps de réponse du serveur est trop long (*sur...*). ~> `glpi_server_pause_timeout`.
- *Update db...* configure la fréquence de la synchronisation de la base de données pendant le téléversement. ~> `glpi_inventory_update_range`.

### 36.6.6 Éléments actuellement envoyé par WAPT à GLPI

TABLEAU 2 – Description des éléments

Valeur	Envoyé	Non Envoyé
Nom de l'ordinateur	✓	
Nom d'utilisateur	✓	
Description	✓	
Nom de l'OS	✓	
Version de l'OS	✓	
Langue	✓	
CPU	✓	
Mémoire	✓	
Batterie	✓	
Type de châssis	✓	
Physique ou virtuel	✓	
Configuration de la carte réseau	✓	
Liste d'imprimante et les propriétés	✓	
Logiciel installé <sup>1</sup>	✓	
Lecteur réseau	✓	
Variables d'environnement <sup>2</sup>	✓	
Modèles des écrans	✓	
Modèle de la souris et du clavier		✗
Modèles des cartes contrôleurs (excepté la carte graphique)		✗
Version de l'antivirus		✗
État du parefeu		✗
Liste groupe local		✗
Liste et état de la banque de mémoire		✗
Liste des ports USB et des périphériques connectés		✗
Statut de l'imprimante		✗
Lecteurs de carte		✗
Liste d'appx à l'échelle du système		✗

1. Sans compter l'installation des Appx à l'échelle du système

2. Actuellement, les variables d'environnement du système et de l'utilisateur sont incluses.

### 36.6.7 Erreur sur l'inventaire

Si l'inventaire n'est pas correct, il est possible que GLPI par WAPT ait des éléments manquants.

Il est possible d'envoyer un rapport à l'équipe de développement de Tranquil IT.

Pour le rapport, l'équipe de développement a besoin de 2 fichiers `.xml`.

1. Sur l'ordinateur qui a besoin d'envoyer l'élément, installez l'agent <https://github.com/fusioninventory/fusioninventory-agent/releases>
2. Obtenez l'inventaire local :

— Window :

```
"C:\Program Files\FusionInventory-Agent\fusioninventory-inventory" > c:\inventory.xml
```

— Linux / MAC :

```
fusioninventory-inventory > inventory.xml
```

3. Définissez le répertoire de débogage dans le fichier `waptserver.ini`

```
glpi_inventory_debug_directory = /tmp/glpi
```

4. Redémarrez le serveur WAPT
5. Récupérez le fichier `UUID.xml`, l'UUID doit être le même que celui de l'ordinateur précédent.
6. Envoyez les 2 fichiers à l'équipe de développement.

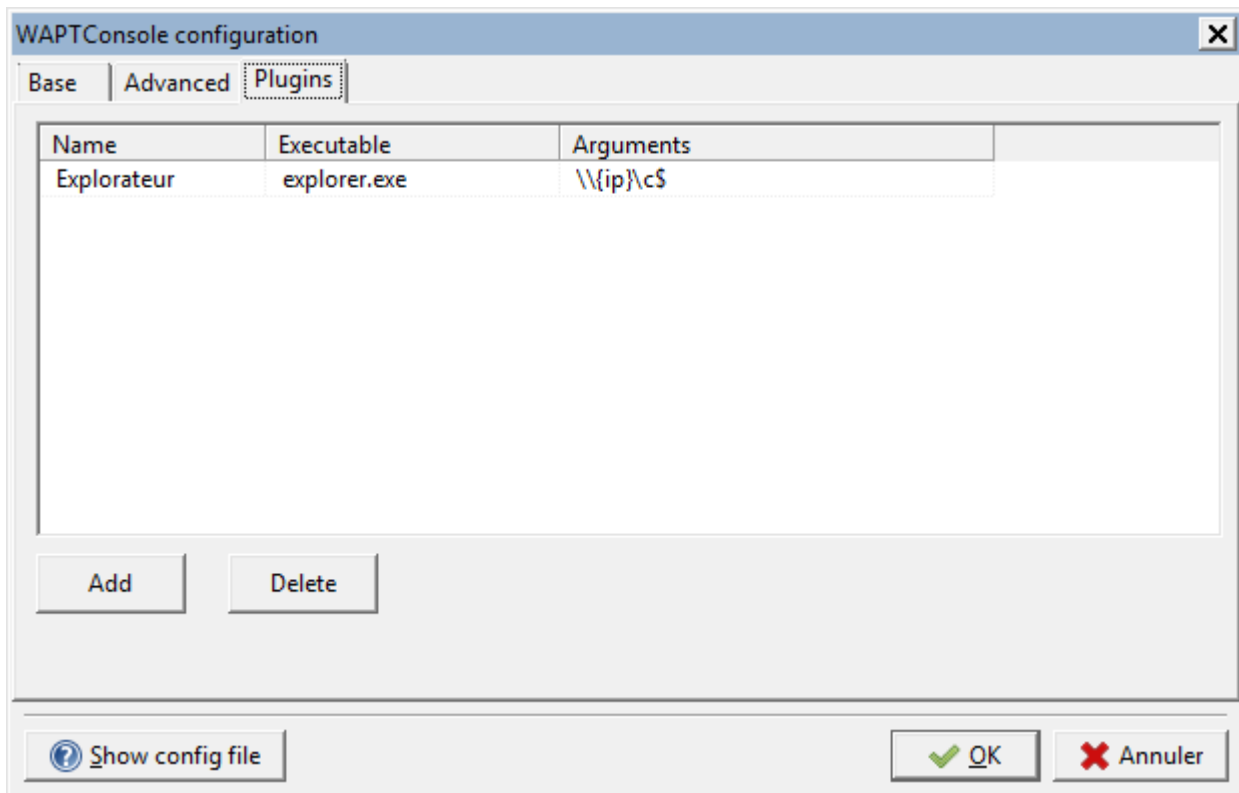
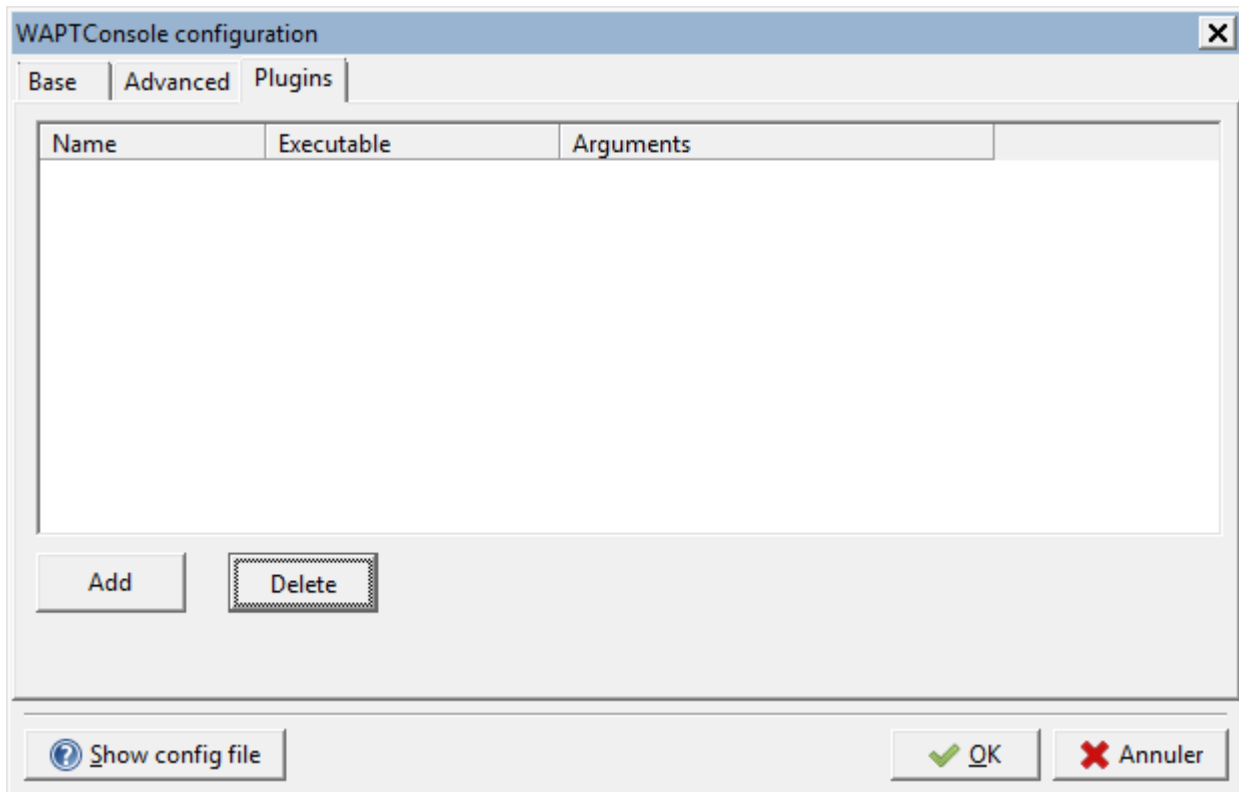
## 36.7 Ajouter des plugins dans la Console

Pour ajouter des plugins, aller dans *Outils*, *Préférences* et onglet *Outils externes*.

Cliquez sur *Ajouter* pour ajouter un plugin, puis éditez les colonnes correspondantes.

Colonne	Description
Nom	Le nom qui apparaîtra dans le menu.
Exécutable	Chemin de l'exécutable qui sera exécuté après le clic.
Arguments	Les arguments passés à l'exécutable. Des variables peuvent être utilisées telles que {ip}, {uuid} ou {computer_fqdn}.

Les plugins vont alors apparaître dans le menu :



## 36.8 Re-Signer tous les paquets avec la console WAPT

Cette méthode pour re-signer tous les paquets hôtes est utile lorsque la méthode cryptographique sous-jacente ou que la librairie change, comme ce cas-ci lors de la mise à jour de WAPT 1.8.2 (basé sur Python 2.7) et WAPT >= 2.0 (basé sur Python 3.x).

---

**Indication :** Utilisez le certificat Administrateur pour re-signer les paquets.

---

### 36.8.1 Les paquets hôtes

- Sélectionnez tous les hôtes.
- Clic-droit sur les hôtes sélectionnés.
- Sélectionnez *Resigner les paquets de configuration machine*.
- Confirmez la re-signature sur les hôtes sélectionnés.
- Puis, entrez votre clé privée.
- Les paquets *hôtes* sélectionnés sont désormais tous re-signés avec la nouvelle méthode cryptographique exigée par Python3.

### 36.8.2 Autre type de paquet WAPT

- Accédez au dépôt depuis votre console WAPT.
- Sélectionnez tous les paquets dans le dépôt, puis clic-droit sur la sélection.
- Sélectionnez *Resigner les paquets*.
- Pour lancer le processus de signature, cliquez sur *Resigner les paquets*.
- Après ce procédé, ce qui prend du temps, tous les paquets seront re-signés.

**Attention :**



microsoft-office

16.0.12325.20276-2

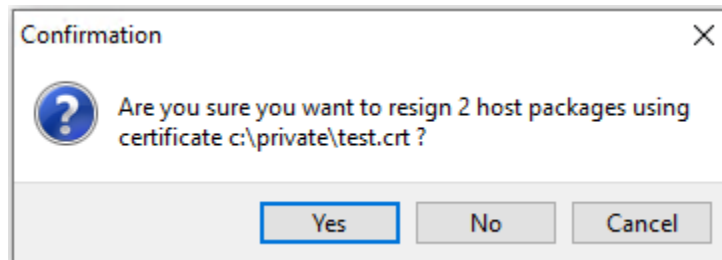
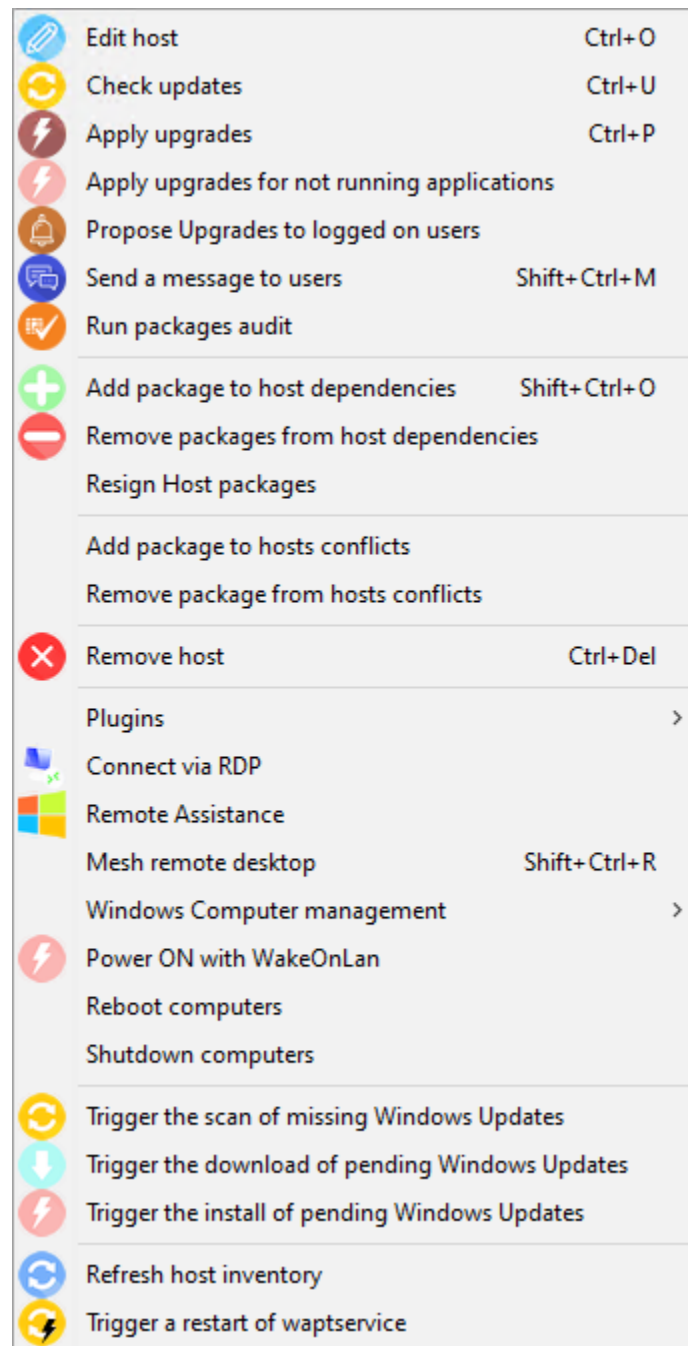
PROD

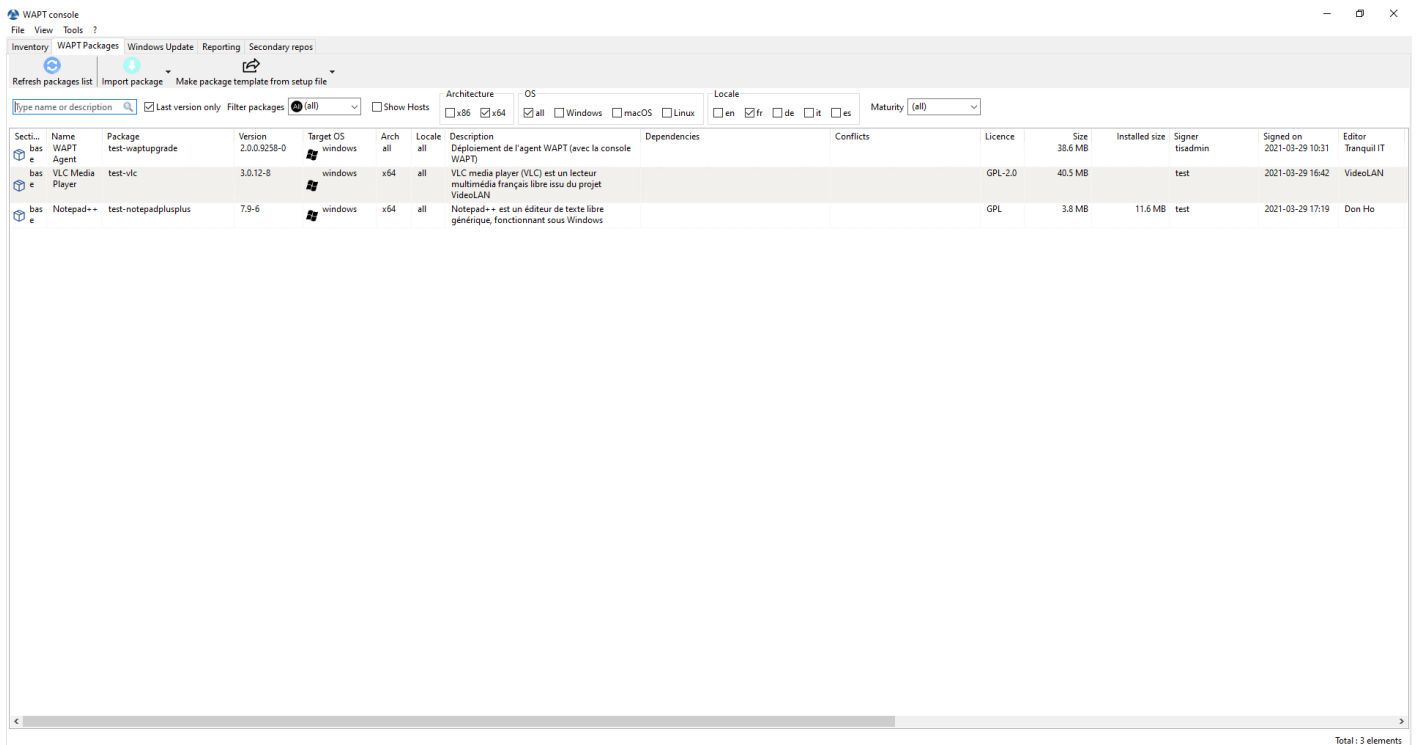
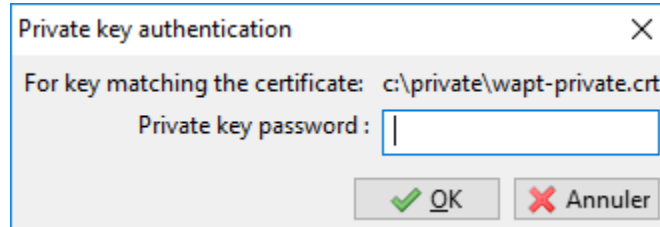
ERROR

Access violation

Si l'erreur **Access violation** apparaît, c'est parce que le paquet est trop gros.

Éditez le paquet et suivez *cette procédure*.

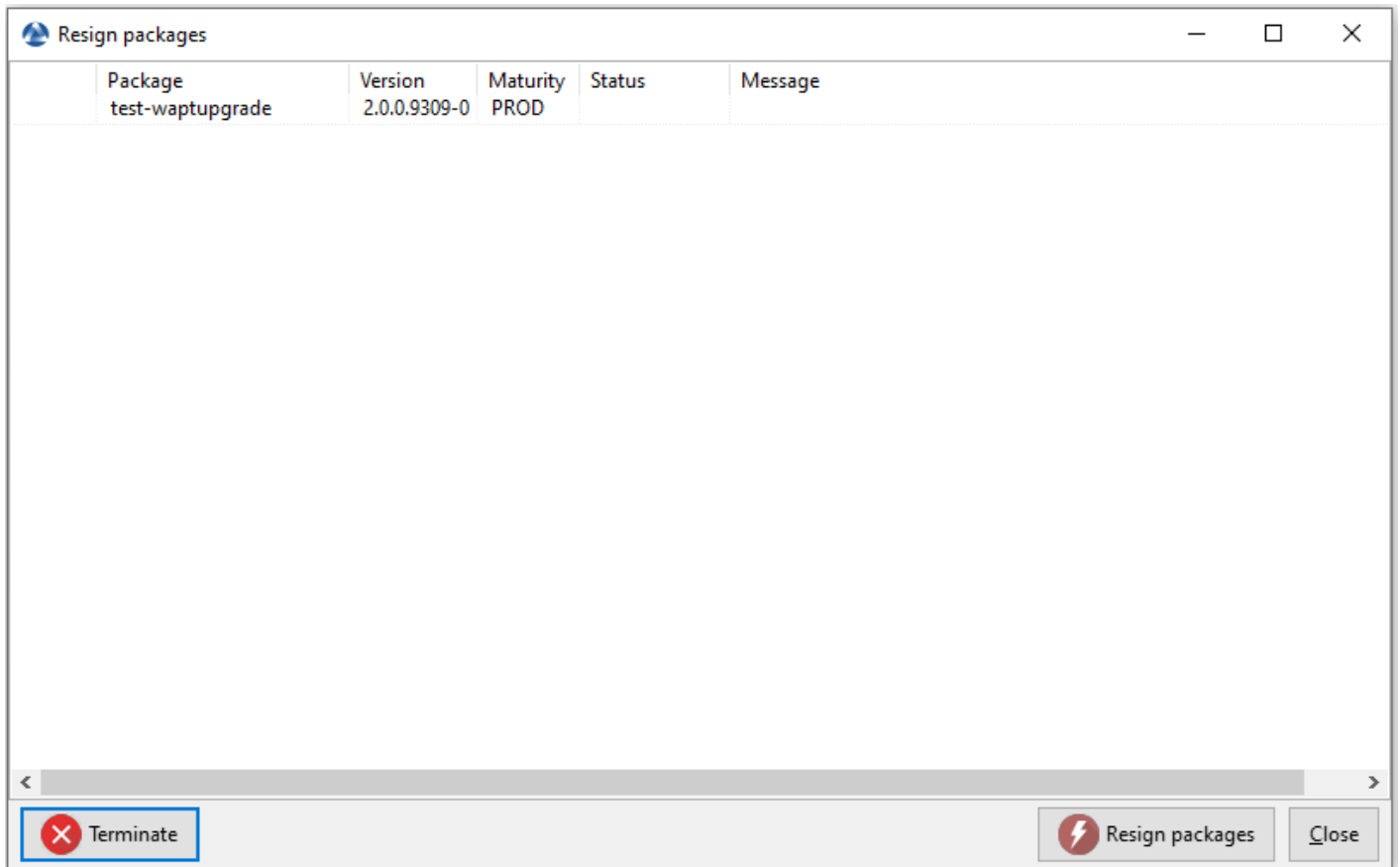
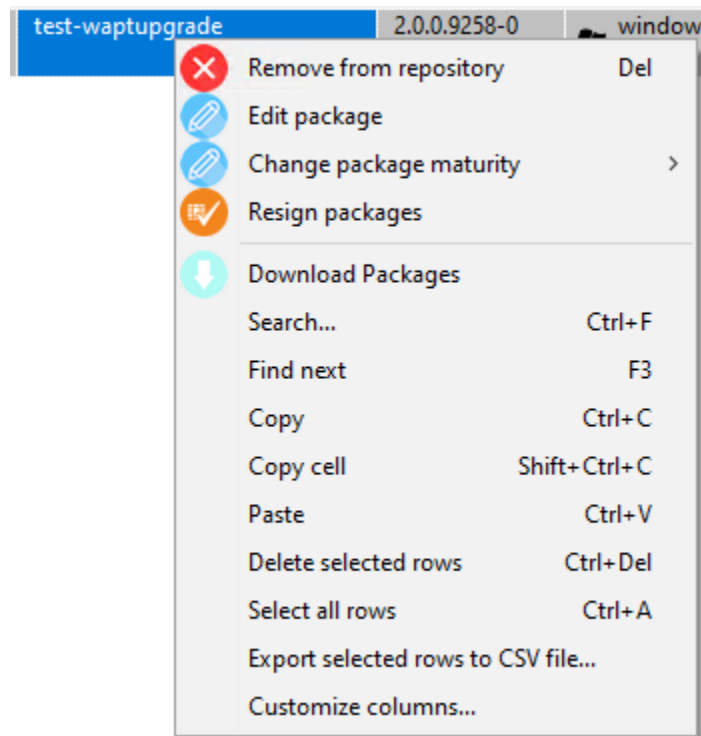





A screenshot of the WAPT console interface. The window title is "WAPT console". The menu bar includes "File", "View", and "Tools". The main interface has a toolbar with "Refresh packages list", "Import package", and "Make package template from setup file". Below the toolbar is a search bar and several filter options: "Last version only", "Filter packages", "Show Hosts", "Architecture" (x86, x64), "OS" (all, Windows, macOS, Linux), "Locale" (en, fr, de, it, es), and "Maturity (all)". The main area displays a table of packages with the following columns: Secteur, Name, Package, Version, Target OS, Arch, Locale, Description, Dependencies, Conflicts, Licence, Size, Installed size, Signer, Signed on, and Editor. The table contains three rows of data.

Secteur	Name	Package	Version	Target OS	Arch	Locale	Description	Dependencies	Conflicts	Licence	Size	Installed size	Signer	Signed on	Editor
bas	WAPT Agent	test-waptupgrade	2.0.0.9258-0	windows	all	all	Déploiement de l'agent WAPT (avec la console WAPT)						tsadmin	2021-03-29 10:31	Tranquil IT
bas	VLC Media Player	test-vlc	3.0.12-8	windows	x64	all	VLC media player (VLC) est un lecteur multimédia français libre issu du projet VideoLAN			GPL-2.0	40.5 MB		test	2021-03-29 16:42	VideoLAN
bas	Notepad++	test-notepadplusplus	7.9-6	windows	x64	all	Notepad++ est un éditeur de texte libre générique, fonctionnant sous Windows			GPL	3.8 MB	11.6 MB	test	2021-03-29 17:19	Don Ho

Total: 3 elements



	Package	Version	Maturity	Status
	test-waptupgrade	2.0.0.9258-0	PROD	OK



---

## Utilisation du Self-Service de WAPT

---

### 37.1 Présentation

Avec WAPT, vos utilisateurs peuvent disposer d'un self-service pour l'installation des logiciels.

C'est différent dans les versions **Discovery** et **Enterprise**.

Fonctionnalité	Discovery	Enterprise
Accès au self-service	✓	✓
Déploiement du paquet de self-service	✓	✓
Filtrage des paquets self-service	✗	✓
Onglet de gestion	✗	✓

### 37.2 Principe de fonctionnement

Les *Utilisateurs* gagnent en autonomie en déployant des logiciels et des configurations qui sont fiable et autorisés par l'*Organisation*. C'est un gain de temps pour le support informatique utilisateur de l'*Organisation*.

#### 37.2.1 Discovery

Seuls les Administrateurs locaux et les membres du group *waptself-service* peuvent accéder au self-service sur l'hôte.

**Attention :** Ces utilisateurs ont accès à tous les paquets de votre dépôt.

## 37.2.2 Enterprise

Vous pouvez filtrer la liste des paquets self-service disponibles pour vos utilisateurs.

Un paquet *self-service* peut être déployé sur les hôtes pour lister les différentes règles self-service à appliquer sur l'hôte.

Les paquets *self-service* sont basés sur des groupes d'utilisateurs.

Vos utilisateurs pourront installer une sélection de paquets WAPT sans avoir besoin d'être un *Administrateur local*.

## 37.3 Utilisation de la fonction self-service

### 37.3.1 Configuration

#### Discovery

Sur Discovery, créez un groupe *self-service* dans votre Active Directory et ajoutez vos utilisateurs. Ces utilisateurs et tous les *Local Administrator* ont accès à tous les paquets de votre dépôt.

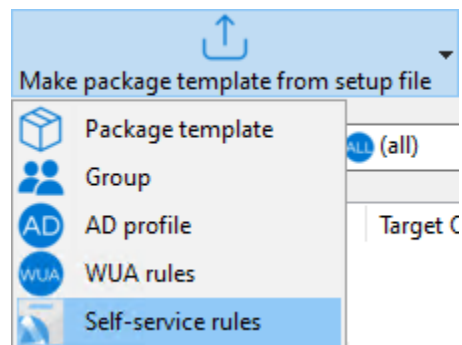
---

**Note :** Il n'est pas possible de filtrer les paquets rendus accessibles à l'utilisateur.

---

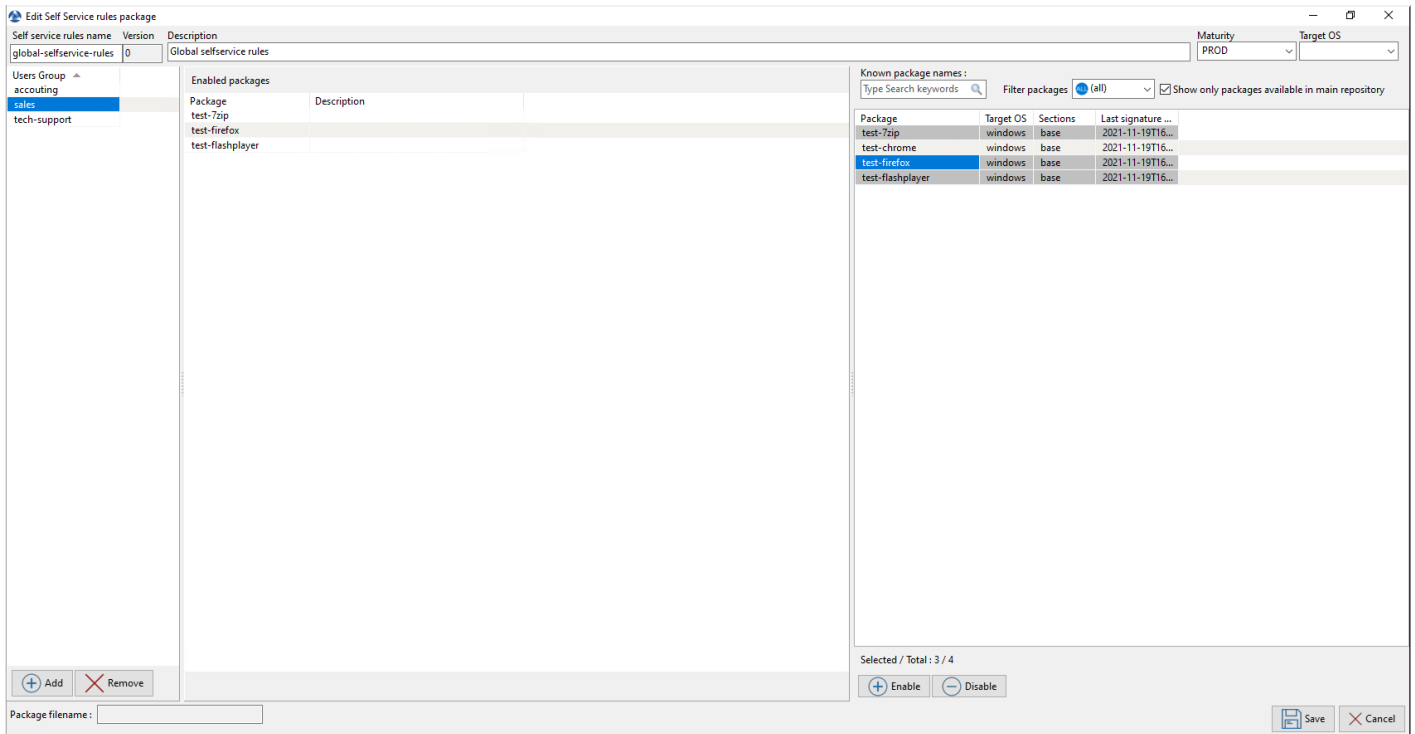
#### Enterprise

Dans la console, allez dans l'onglet *Dépôt privé* et créez *Règles de self-service*.



Vous pouvez désormais créer votre premier paquet de règle *self-service*.

1. Donnez un nom à votre paquet *self-service* ;
2. Donner une description ;
3. Cliquez sur *Ajouter* pour ajouter un groupe Active Directory (en bas à gauche) ;
4. Nommez le groupe *self-service* (avec F2 ou tapez directement dans la cellule). Le groupe doit avoir le même nom que le **groupe de sécurité des utilisateurs de l'Active Directory** ;
5. Faites glisser et déposez dans la colonne centrale les logiciels et les paquets de configuration autorisés pour ce groupe *de libre-service* ;
6. Ajoutez autant de groupes que vous le souhaitez dans le paquet ;
7. Sauvegardez le paquet et déployez le paquet sur votre sélection d'hôtes ;

**Note :**

- Si un groupe apparaît dans plusieurs paquets *self-service*, alors les règles sont fusionnées ;
- L'authentification utilisée est l'authentification système par défaut, il est possible de s'authentifier avec *Active Directory*.
- Une fois le paquet déployé, seuls les paquets autorisés figurant dans le(s) groupe(s) *self-service* dont l'*Utilisateur* est membre seront affichés à l'*Utilisateur* connecté ;

## 37.4 Utilisation de l'application Self-Service

Le self-service est accessible dans le menu de démarrage sous le nom *Self-Service logiciel WAPT* :

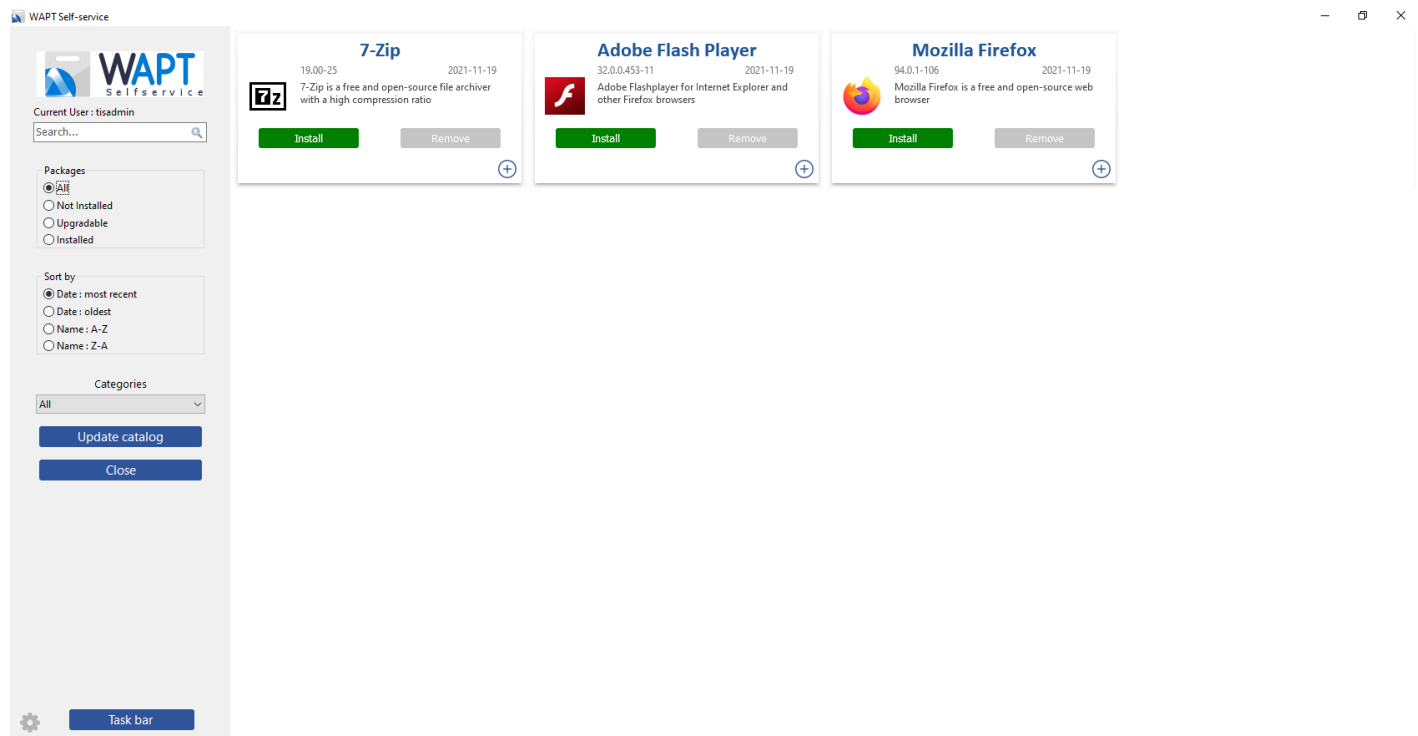



Il est aussi disponible directement dans le dossier de WAPT `<base>\waptself.exe`.

**Note :** L'identifiant et le mot de passer à entrer lors du lancement du self-service sont ceux de l'*Utilisateur* (local ou Active directory).

Le self-service affiche alors une liste de paquets disponibles pour l'installation.

- L'utilisateur peut avoir plus de détails sur chaque paquet avec l'icône + ;
- Différents filtres sont disponibles pour l'utilisateur sur le panneau de gauche ;



- Le bouton *Mettre à jour le catalogue* est utilisé pour forcer un **wapt-get update** sur l'agent WAPT ;
- La liste des catégorie de paquet est affiché pour l'utilisateur.
- La liste des tâches en cours de l'agent WAPT est disponible avec le bouton *Barre de tâches* ;
- Il est possible de changer la langue de l'interface avec le bouton  en bas à gauche.


### 37.4.1 Catégories de paquets disponibles par défaut

Par défaut, WAPT gère ces catégories de paquets :

- Internet ;
- Utilitaires ;
- Messagerie ;
- Securite ;
- Systeme et reseau ;
- Stockage ;
- Media ;
- Developpement ;
- Bureautique ;

Il est possible de ajouter d'autres catégories aux paquets que vous développez.

[Hide](#)

 **Google Chrome**

**Editor :** Google  
**Last version :** 96.0.4664.45-30  
**Official website :** <https://www.google.com/chrome/>  
**Licence :** Freeware  
**Impacted process :**  
chrome, GoogleUpdate, chrome\_proxy, GoogleCrashHandler64, Goc  
**Maintainer :** WAPT Team, Tranquil IT, Jimmy PELÉ, Simon FONTENE  
**Package size :** 80592 kB ( 82526084 bytes )  
**Size installed :**  
**Repository :** wapt  
**Signer :** documentation  
**Package name :** test-chrome  
**Dependency :**  
**Conflicts :**  
**Section :** base  
**Signature Date :** 11/19/2021 16:16:46  
**Categories :** Internet

### Description

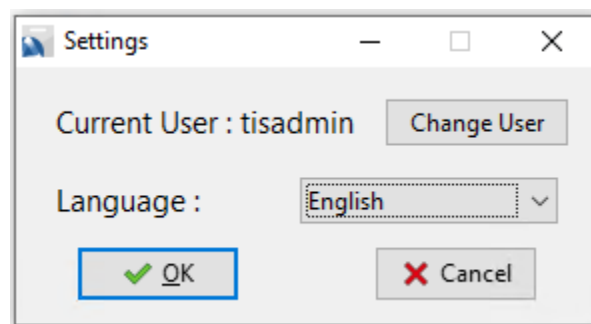
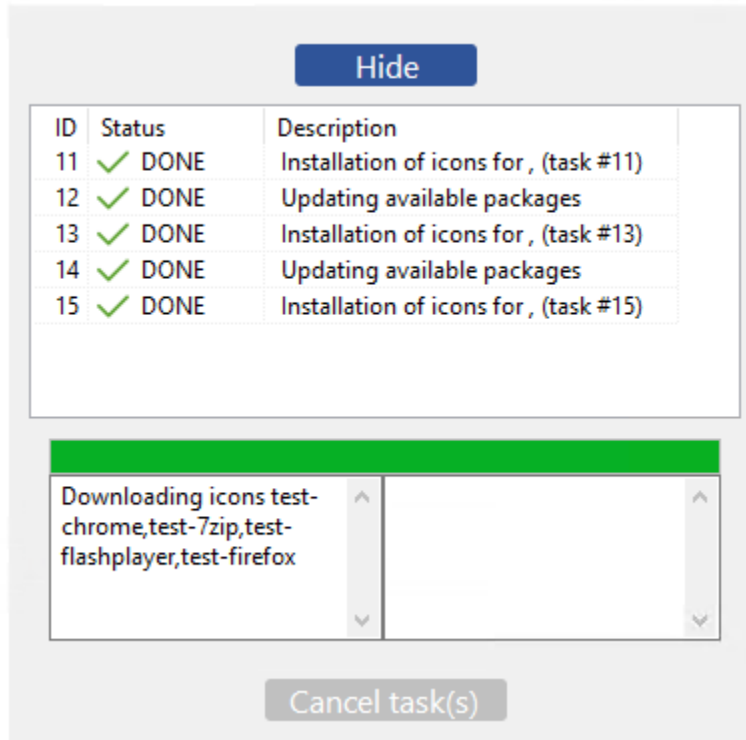
Google Chrome is a web browser created by Google and based on the open source project Chromium

Packages

All  
 Not Installed  
 Upgradable  
 Installed

Sort by

Date : most recent  
 Date : oldest  
 Name : A-Z  
 Name : Z-A



## 37.5 Les configurations de l'agent WAPT pour le WAPT Self-Service

L'agent WAPT peut être configuré pour le self-service de WAPT.

### 37.5.1 Configurer une méthode d'authentification différente pour le self-service

Par défaut, l'authentification sur le service WAPT est configurée en mode système.

Ce comportement est défini avec la valeur de `service_auth_type` dans `wapt-get.ini` :

Valeur	Description
<code>système</code> <i>Valeur par défaut</i>	Le service WAPT transmet l'authentification directement au système d'exploitation; il récupère également les groupes en interrogeant directement le système d'exploitation.
<code>waptserver-ldap</code>	Ce mode permet l'authentification auprès du serveur WAPT. Le serveur WAPT fera une requête LDAP pour vérifier l'authentification et les groupes. Pour que cela fonctionne, vous devez avoir configuré <i>l'authentification LDAP</i> sur le serveur WAPT.
<code>waptagent-ldap</code>	Ce mode permet l'authentification avec un serveur LDAP identifié dans <code>wapt-get.ini</code> . L'agent WAPT fera une requête LDAP pour vérifier l'authentification et les groupes. Pour que cela fonctionne, vous devez avoir configuré <i>l'authentification LDAP</i> sur le serveur WAPT.

Vous pouvez consulter cet article décrivant les *paramètres pour le Self-Service de WAPT et l'Authentification du Waptservice* pour plus d'options.

**Note :** Pour que l'authentification système sous GNU/Linux fonctionne correctement, assurez-vous de correctement configurer l'authentification pam et votre `nsswitch.conf`. La commande `id username` doit renvoyer la liste des groupes dont l'utilisateur est membre.

**Avertissement :** En mode `system` nous supposons que les *Administrateurs Locaux* peut voir tous les paquets. Pour changer ce comportement, passez au point suivant.

### 37.5.2 Configuration de l'authentification pour Administrateur

Par défaut le Self-Service WAPT utilise l'authentification `system`.

Dans ce mode, les *Administrateurs Locaux* peuvent voir tous les paquets.

Si vous ne voulez pas de ce comportement vous avez 2 possibilités :

- Bloquer l'affichage de tous les paquets pour les *Administrateurs Locaux*
- Tous les paquets ne sont visible que pour un groupe d'utilisateurs spécifique

### Bloquer les Administrateurs Locaux sur le self-service

Pour bloquer l'affichage de tous les paquets pour les *Administrateurs Locaux* vous devez ajouter le paramètre `waptservice_admin_filter` dans `wapt-get.ini`.

Valeur	<i>True</i>	<i>False</i>
<code>waptservice_admin_filter</code>	Active le filtrage d'affichage du <i>paquet selfservice</i> pour les administrateurs locaux.	Désactiver le filtrage d'affichage du <i>paquet selfservice</i> pour les administrateurs locaux.

### Groupe d'utilisateurs Administrateur du self-service

Il est possible d'utiliser un groupe d'utilisateurs spécial pour définir une liste d'administrateurs dans le Self-Service.

Créez un groupe de sécurité d'utilisateurs nommé « `waptselfservice` » et ajoutez des membres.

Tous les membres de ce groupe peuvent voir tous les paquets sur le Self-Service WAPT.

Avec le paramètre `waptservice_admin_filter`, vous avez sécurisé l'accès administrateur de WAPT Self-Service.

## 37.6 Vidéo de démonstration

[https://youtu.be/-\\_sm8KBwDOW](https://youtu.be/-_sm8KBwDOW)



## Utiliser le WAPTtray

WAPTtray est un programme systray. Il fonctionne dans le contexte de l'utilisateur.

WAPTtray se lance à l'ouverture de session si l'option a été cochée lors de l'installation de l'agent WAPT. L'icône apparaîtra dans la barre d'outils de la zone de notification de Windows.

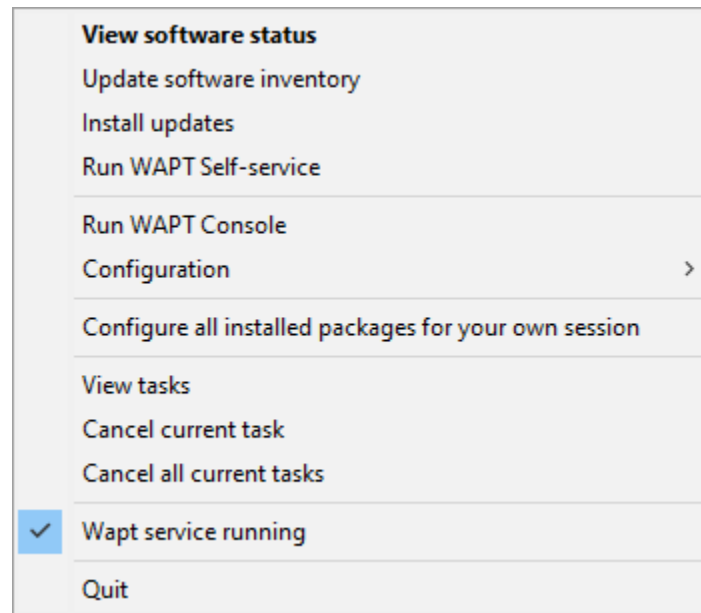
On peut aussi lancer WAPTtray manuellement sur C:\Program Files (x86)\wapt\wapttray.exe.

### 38.1 Les fonctionnalités du WAPTtray

#### 38.1.1 Fonctions principales

TABLEAU 1 – Liste des fonctionnalités de WAPTtray

Action	Description
Afficher le statut des logiciels	Lance l'interface web local dans un navigateur
Mettre à jour de l'inventaire des logiciels	Rafraîchir la liste de paquet disponibles. Double-clic sur l'icône fait la même action.
Installer les mises à jour	Lance l'installation d'une mise à jour en attente
Lancer le Self-service	Lance le Self-Service de WAPT
Lancer la console WAPT	Lance la console WAPT
Configuration	Voir le tableau suivant pour les options détaillées
Configurer tous les paquets installés pour votre session	Lance un <b>session-setup</b> pour configurer tous les paquets installés dans l'environnement utilisateur
Affiches les tâches	Afficher la liste des tâches sur l'interface web locale dans le navigateur
Annuler la tâche en cours	Annule la tâche en cours d'exécution sur l'agent WAPT
Annuler toutes les tâches en cours	Annule toutes les tâches en cours d'exécution sur l'agent WAPT
Service WAPT lancé	Arrête et relance le <i>WAPTservice</i>
Quitter	Ferme l'icône sans stopper le <i>WAPTservice</i> local



### 38.1.2 Fonctions de configuration

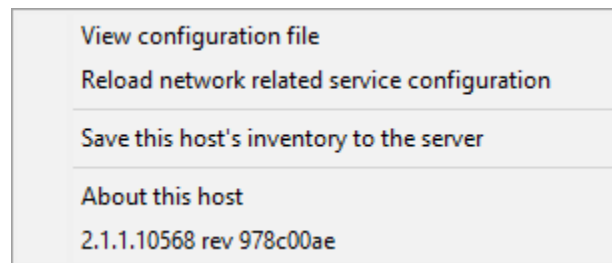


TABLEAU 2 – Liste des configurations

Action	Description
Afficher le fichier de configuration	Ouvre le fichier C:\Program Files (x86)wapt\wapt-get.ini avec les privilèges <i>Administrateur local</i> (les informations d'identification peuvent être demandées)
Recharger la configuration réseau du service	Relance la connexion au serveur WAPT en cas de reconfiguration du réseau
Sauvegarder l'inventaire de cette machine sur le serveur	Mise à jour de l'inventaire de l'hôte avec le serveur WAPT
Informations de la machine	Lance l'interface web locale dans un fichier de navigation avec les privilèges d' <i>Administrateur Local</i> (les informations d'identification peuvent être demandées) pour afficher l'inventaire des hôtes

## 38.2 Vidéo de démonstration

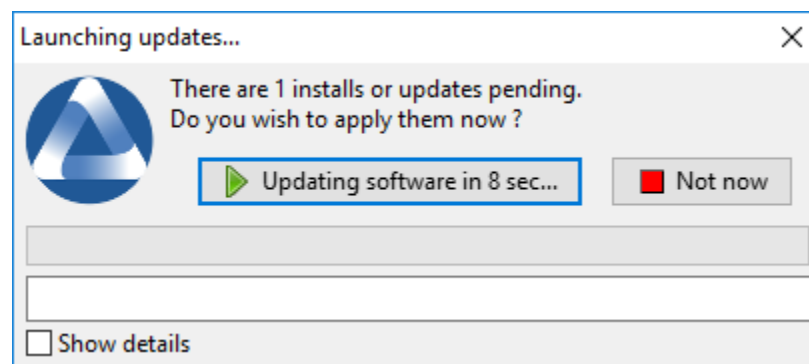
<https://youtu.be/9iG36IeHuVc>



WAPTEExit permet de mettre à jour et d'installer les paquets WAPT lorsqu'un hôte s'arrête, à la demande de l'utilisateur, ou à une heure programmée.

Le mécanisme est simple. Si les paquets sont en attente de mise à jour, ils seront installés.

**Indication :** La méthode WAPTEExit est très efficace dans la plupart des situations car elle ne nécessite pas l'intervention du *User* ou du *Administrator*.



WAPTEExit s'exécute par défaut à l'arrêt, il est installé avec l'agent WAPT.

Le comportement de WAPTEExit est personnalisable dans *wapt-get.ini*.

**Avertissement :** Si une tâche est en cours d'exécution, l'arrêt est suspendu jusqu'à ce que la tâche soit terminée.

## 39.1 Déclencher manuellement l'exécution de WAPTextit

On peut aussi lancer WAPTtray manuellement sur C:\Program Files (x86)\wapt\waptexit.exe.

## 39.2 Déclencher WAPTextit avec une tâche planifiée

On peut déployer une GPO ou un paquet WAPT qui va déclencher le WAPTextit à un moment prédéfini.

**Déclencher WAPTextit avec une tâche planifiée convient mieux aux serveurs qui ne sont pas arrêtés fréquemment.**

Vous pouvez adapter *la procédure de déploiement de l'agent WAPT* pour déclencher le script WAPTextit.exe au moment de votre choix.

---

**Indication :** Vous pouvez utiliser le script suivant pour votre tâche planifiée, adaptée à vos besoins :

```
waptpython -c "from waptservice.enterprise import start_waptexit start_waptexit('', {'only_priorities
↳ :False, 'only_if_not_process_running':True, 'install_wua_updates':False, 'countdown':300}, 'schtask
↳ ')"
```

### Avertissement :

- Tous les logiciels en cours d'exécution qui sont mis à jour peuvent être détruits avec une possible perte de données.
- WAPTextit peut échouer à mettre à niveau un logiciel si un logiciel que vous mettez à niveau figure dans la liste `impacted_process` du fichier `:file :control`. Voir *sous* pour plus d'informations.
- La méthode consistant à déclencher WAPTextit à une heure planifiée est la moins recommandée pour les ordinateurs de bureau. Il est préférable de laisser le WAPTextit s'exécuter à l'arrêt ou à la demande de l'utilisateur.

## 39.3 Paramètres de WAPTextit dans wapt-get.ini

Il est possible de *modifier le comportement de WAPTextit* dans le `wapt-get.ini`.

Il est également possible de modifier le comportement de WAPTextit directement depuis la ligne de commande, voir les points suivants.

## 39.4 Paramètres de WAPTextit dans le binaire

### 39.4.1 Empêcher l'annulation des mises à jour

Pour désactiver l'interruption de l'installation des mises à jour, vous pouvez exécuter WAPTextit avec l'argument :

```
waptexit.exe -allow_cancel_upgrade = True
```

### 39.4.2 Augmenter le temps de déclenchement dans waptexit

Pour spécifier le temps d'attente avant le démarrage automatique des installations, vous pouvez lancer WAPTEExit avec l'argument :

```
waptexit.exe -waptexit_countdown = 10000
```

### 39.5 Ne pas interrompre l'activité de l'utilisateur

Pour dire à WAPT de ne pas exécuter une **upgrade** du logiciel en cours d'exécution sur la machine (attribut *impacted\_process* du paquet), vous pouvez exécuter **waptexit** avec l'argument :

```
waptexit.exe -only_if_not_process_running=True
```

Sinon, **waptexit** prendra la valeur indiquée dans C:\Program Files (x86)wapt\wapt-get.ini :

### 39.6 Lancement de l'installation de paquets avec un niveau de priorité spécial

Pour dire à WAPT de ne mettre à jour qu'un paquet spécifique priority, vous pouvez exécuter **waptexit** avec l'argument :

```
waptexit.exe -priorities = high
```

### 39.7 Activer/désactiver WAPTexit

Pour activer ou désactiver **waptexit** dans les scripts de stratégie de groupe locale, utilisez :

— pour activer le **waptexit** à l'extinction du poste :

```
wapt-get add-upgrade-shutdown
```

— pour désactiver le **waptexit** à l'extinction du poste :

```
wapt-get remove-upgrade-shutdown
```

### 39.8 Vidéo de démonstration





Il est possible de personnaliser WAPT aux couleurs de votre société.

3 programmes sont personnalisables :

- WAPTExit
- WAPT Self-Service
- WAPT Message

Il est possible d'utiliser le même logo pour tous les programmes.

Placer l'image dans `<wapt_folder>\templates`.

Le logo doit être nommé `wapt-logo.png`

---

**Note :** La taille recommandée pour le logo est 200X55 et le format est `.png`

---

Pour différents logo par programme, voir les points suivants.

### 40.1 WAPTExit

Il est possible de personnaliser waptexit en plaçant l'image que vous voulez dans `<wapt_folder>\templates`

Le logo doit être nommé `waptexit-logo.png`

---

**Note :** La taille recommandée pour le logo est 200X55 et le format est `.png`

---

**Avertissement :** S'il n'est pas défini, WAPT utilise `wapt-logo.png`. S'il n'existe pas, utilisez un logo WAPT par défaut.

## 40.2 WAPT Self-Service

Il est possible de personnaliser waptexit en plaçant l'image que vous voulez dans `<wapt_folder>\templates`

Le logo doit être nommé `waptself-logo.png`

---

**Note :** La taille recommandée pour le logo est 200X55 et le format est `.png`

---

**Avertissement :** S'il n'est pas défini, WAPT utilise dans l'ordre `waptexit-logo.png`, `waptself-logo.png` et enfin le logo WAPT par défaut.

## 40.3 WAPT Message

Il est possible de personnaliser waptexit en plaçant l'image que vous voulez dans `<wapt_folder>\templates`

Le logo doit être nommé `waptmessage-logo.png`

---

**Note :** La taille recommandée pour le logo est 200X55 et le format est `.png`

---

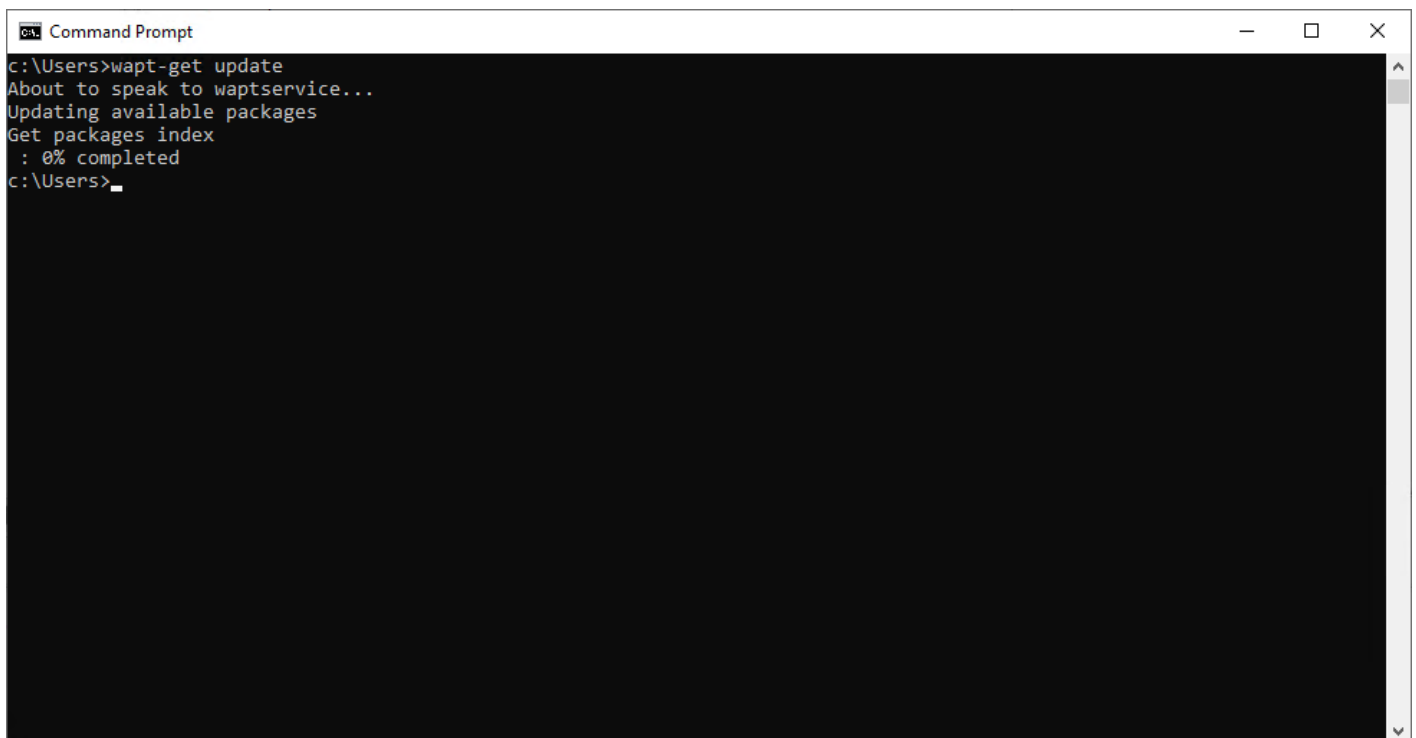
**Avertissement :** S'il n'est pas défini, WAPT utilise dans l'ordre `waptexit-logo.png`, `waptself-logo.png` et enfin le logo WAPT par défaut.

---

## Utiliser WAPT en ligne de commande

---

L'agent WAPT fournit un utilitaire d'interface de ligne de commande **wapt-get**.



```
Command Prompt
c:\Users>wapt-get update
About to speak to waptservice...
Updating available packages
Get packages index
: 0% completed
c:\Users>
```

FIG. 1 – L'invite de commande Windows

---

**Note :**

- Par défaut, les actions en ligne de commande dans WAPT sont exécutées avec les droits de l'utilisateur qui a lancé le **cmd.exe**.
  - Si le **cmd.exe** n'a pas été lancé avec les privilèges de *Local Administrator*, la commande sera transmise au **waptservice**.
  - Par sécurité, certaines actions demandent un identifiant et un mot de passe.
  - Seuls les *Administrateurs Locaux* et les membres du groupe de sécurité Active Directory *waptselfservice* sont autorisés.
  - Pour forcer l'utilisation du service WAPT en tant qu'*Administrateur Local*, ajouter simplement **-S** après **wapt-get.exe**.
- 

## 41.1 Utilisation des fonctions les plus courantes dans WAPT via ligne de commande

### 41.1.1 wapt-get install

La commande `wapt-get install <package name>` lance l'installation d'un paquet.

La commande prend un argument. Cet argument est le nom paquet ou une liste de noms de paquets avec le préfixe du référentiel.

**Indication :** Il est possible d'installer plusieurs paquets avec la même commande :

```
wapt-get install package1 package2
```

---

**Note :** Si le paquet n'a pas été téléchargé dans le cache, `install` va d'abord télécharger le paquet dans le cache, puis l'installer.

---

**Attention :** L'installation d'un paquet WAPT avec `install` n'ajoute pas le paquet comme dépendance à l'hôte.

Le paquet est installé sur la machine, mais si l'ordinateur est réimagé, le paquet ne sera pas réinstallé automatiquement.

La commande `wapt-get install tis-firefox` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
installing WAPT packages tis-firefox
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 14121562 /_
↪54313787 (26%) (24624 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 33131357 /_
↪54313787 (61%) (29414 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 50511741 /_
↪54313787 (93%) (30412 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 54313787 /_
↪54313787 (100%) (30360 KB/s)
Installing tis-firefox(=94.0.1-106)
Installing: Firefox_Setup_94.0.1.exe
Waiting for key key Mozilla Firefox 94.0.1 (x64 en-US) to appear in Windows registry
```

(suite sur la page suivante)

(suite de la page précédente)

```
Delete C:\Program Files (x86)\wapt\cache\tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt
```

Results :

```
=== install packages ===
  tis-firefox [x64_en_PROD]      | tis-firefox (94.0.1-106)          | tis-firefox (50.0.2-
↪73)
```

## 41.1.2 wapt-get update

La commande `wapt-get update` permet de mettre à jour la liste des paquets disponibles.

L'agent WAPT local téléchargera le fichier Packages du dépôt privé et le comparera à sa base de données locale.

- Si de nouvelles mises à jour sont disponibles, l'agent WAPT fait passer le statut des paquets à **TO-UPGRADE**.
- Si de nouveaux logiciels ont été ajoutés sur le dépôt, ils deviennent téléchargeables pas l'agent WAPT.

**Note :** La commande `wapt-get update` ne télécharge pas les paquets, elle met seulement à jour la base de données locale des paquets.

La commande `wapt-get update` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Update package list from https://srvwapt.mydomain.lan/wapt, https://srvwapt.mydomain.lan/wapt-host
Total packages : 8
Added packages :

Removed packages :

Discarded packages count : 6
Pending operations :
  install:
  upgrade:
  additional:
  remove:
  immediate_installs:
Repositories URL :
  https://srvwapt.mydomain.lan/wapt
  https://srvwapt.mydomain.lan/wapt-host
```

### 41.1.3 wapt-get upgrade

La commande `wapt-get upgrade` permet de lancer l'installation des paquets en attente de mise à jour ou en attente d'installation. L'agent WAPT local télécharge si nécessaire les paquets WAPT dans son cache local puis les installe.

---

**Indication :** Il est recommandé de lancer la commande `wapt-get update` avant de lancer une commande `wapt-get upgrade` ; Sans lancement préalable d'un `update`, l'agent WAPT n'installera rien ;

---

La commande `wapt-get upgrade` renvoie :

```
Installing tis-mumble
Shutting down Mumble
installing Mumble 1.2.3

=== install packages ===
tis-mumble
```

### 41.1.4 wapt-get remove

La commande `wapt-get remove <nom du paquet>` supprime un paquet.

La commande prend un argument. Cet argument est le nom paquet ou une liste de noms de paquets avec le préfixe du référentiel.

**Attention :** La suppression d'un paquet WAPT avec `remove` ne supprime pas la dépendance du paquet sur l'hôte.

**Le paquet sera effectivement désinstallé de la machine, mais il sera automatiquement réinstallé lors de la prochaine mise à niveau**

Pour supprimer complètement un paquet d'un hôte, faites un `remove` pour le paquet ciblé, puis modifiez la configuration de l'hôte via la console WAPT pour supprimer la dépendance du paquet sur l'hôte.

La commande `wapt-get remove tis-firefox` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Removing tis-firefox ...

Waiting for the removal of key key Mozilla Firefox 94.0.1 (x64 en-US) from Windows registry
=== Removed packages ===
tis-firefox
```

### 41.1.5 wapt-get uninstall

La commande `wapt-get uninstall <nom du paquet>` exécute la fonction de désinstallation du paquet si elle existe.

La commande prend un argument. Cet argument est le nom paquet ou une liste de noms de paquets avec le préfixe du référentiel.

**Attention :** L'exécution de la fonction de désinstallation du paquet ne supprime pas le paquet sur l'hôte.

La commande `wapt-get uninstall tis-awcleaner` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Uninstalling tis-awcleaner ...
None
Uninstallation done
```

### 41.1.6 wapt-get forget

La commande `wapt-get forget <package name>` installe l'état du paquet à partir de la base de données locale.

La commande prend un argument. Cet argument est le nom paquet ou une liste de noms de paquets avec le préfixe du référentiel.

**Attention :** Oublier le paquet ne désinstalle pas le paquet.

La commande `wapt-get forget tis-awcleaner` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
=== Packages removed from status ===
tis-awcleaner
```

### 41.1.7 wapt-get audit

La commande `wapt-get audit [<nom du paquet>]` exécute la fonction d'audit.

La commande prend un argument. Cet argument est le nom paquet ou une liste de noms de paquets avec le préfixe du référentiel.

Si l'argument est défini, la commande `wapt-get audit [<nom du paquet>]` exécute la fonction d'audit de paquet.

De même, la commande `wapt-get audit` exécute la fonction d'audit pour tous les paquets installés sur l'hôte.

La commande `wapt-get uninstall tis-firefox` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Auditing tis-firefox ...
Auditing tis-firefox
OK: Uninstall Key Mozilla Firefox 94.0.1 (x64 en-US) in Windows Registry.
tis-firefox -> OK
```

### 41.1.8 wapt-get show

La commande `wapt-get show <nom du paquet>` affiche les informations stockées dans le fichier d'index Packages.

La commande prend un argument. Cet argument est le nom paquet ou une liste de noms de paquets avec le préfixe du référentiel.

Si plusieurs versions d'un paquet sont disponibles sur le référentiel, chaque version du paquet sera affichée.

La commande `wapt-get show tis-7zip` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Display package control data for tis-7zip

package      : tis-7zip
version      : 19.00-25
architecture : x64
section      : base
priority     : optional
name         : 7-Zip
categories   : Utilities
maintainer   : WAPT Team,Tranquil IT,Jimmy PELÉ
description  : 7-Zip is a free and open-source file archiver with a high compression ratio
depends       :
conflicts    :
maturity     : PROD
locale       : all
target_os    : windows
min_wapt_version : 1.7
sources      : https://www.7-zip.org/download.html
installed_size :
impacted_process : 7zFM,7z,7zG
description_fr : 7-Zip est un logiciel gratuit et open source d'archivage de fichiers avec un
↳taux de compression élevé
description_pl :
description_de : 7-Zip ist ein Datenkompressionsprogramm mit einer hohen Kompressionsrate
description_es : 7-Zip es un archivador de ficheros con una alta relación de compresión
description_pt : 0 7-Zip é um compactador de arquivos com alta taxa de compressão
description_it :
description_nl :
description_ru : 7-Zip
audit_schedule :
editor        : Igor Pavlov
keywords      : 7zip,7,zip,7-zip,file,archiver,high,compression,ratio
licence       : LGPL
homepage      : https://www.7-zip.org/
package_uid   : dc66ccd1-d987-482e-b792-04e89a3803f7
valid_from    :
valid_until   :
forced_install_on :
changelog     : https://www.7-zip.org/history.txt
min_os_version : 5.0
max_os_version :
icon_sha256sum : eddc038d3625902b6ddeaabd13dd91529e8d457ffbd0c554f96d343ae243a67a
```

(suite sur la page suivante)



(suite de la page précédente)

```

signer          : documentation
signer_fingerprint: 3f2c0a02231a36eafa1f67905f5c083e4b66cb59942f69cbd231d778a1a25b3d
signature       : QzhPeZFrRbjcGzfzqRpoWsDP9Plaz6BBVlL3adq/MRM19D61+Aez/
↳ JiA8skriCgwSErJXbxOPfxusVqqIpEtyoqh/RlRcnmgCQqk2Fig4gmxpz0rHKokukPQlRk+HdC/
↳ uByxSjfp9oXuB3PVG2PZAFifjVbtjEX2QmV+OY6NdMI9dtkxCsn1Xotn2qhu2bwbJWQ0s51rD9emWuQR7l/
↳ 8WXL+HoquuRho4aCeAOYd6Nta9ktVSR2FM6005ZeUOg4fsnMg+hwp2MlD0mBHX37aJm3hLYkGP2xWjpl9YDDxI7ruRXSHyT7YmbILrS0h1m
signature_date  : 2021-11-19T16:15:42.019196
signed_attributes : package,version,architecture,section,priority,name,categories,maintainer,
↳ description,depends,conflicts,maturity,locale,target_os,min_wapt_version,sources,installed_size,
↳ impacted_process,description_fr,description_pl,description_de,description_es,description_pt,
↳ description_it,description_nl,description_ru,audit_schedule,editor,keywords,licence,homepage,
↳ package_uid,valid_from,valid_until,forced_install_on,changelog,min_os_version,max_os_version,
↳ icon_sha256sum,signer,signer_fingerprint,signature_date,signed_attributes
filename        : tis-7zip_19.00-25_x64_windows_0f4137ed1502b5045d6083aa258b5c42_5.0_PROD_
↳ a10c57d7848cf7b145d6cd64bf4d5389.wapt
size            : 1704227
md5sum         : a10c57d7848cf7b145d6cd64bf4d5389

OK Package control signature checked properly by certificate documentation (fingerprint:
↳ 3f2c0a02231a36eafa1f67905f5c083e4b66cb59942f69cbd231d778a1a25b3d )

```

**Note :** Il est possible qu'un message d'avertissement soit affiché avec cette commande, par exemple :

```

WARNING: control data signature can not be validated with certificates [<SSLCertificate cn=
↳ 'documentation' fingerprint=3f2c0a issuer='documentation' validity=2021-11-19 - 2031-11-17 Code-
↳ Signing=True CA=True>]

```

Ceci est normal si votre certificat n'est pas fiable. Seul le fichier `control` est disponible pour `wapt-get show` et non pour tous les paquets.

Si vous voulez vérifier le paquet correctement, téléchargez-le dans le cache et exécutez la commande `wapt-get show` sur le paquet local.

Par exemple :

```

wapt-get download tis-7zip
wapt-get show "C:\Program Files (x86)\wapt\cache\tis-7zip_19.00-25_x64_windows_
↳ 0f4137ed1502b5045d6083aa258b5c42_5.0_PROD_a10c57d7848cf7b145d6cd64bf4d5389.wapt"

```

### 41.1.9 wapt-get show-params

La commande `wapt-get show-params <nom du paquet>` renvoie des listes de *paramètres* qui seraient passés à la commande `wapt-get install <nom du paquet> --params=PARAMS`.

La commande prend un argument. Cet argument est le nom paquet ou une liste de noms de paquets avec le préfixe du référentiel.

La commande `wapt-get show tis-7zip` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
tis-7zip : {True, 'documentation': True}
```

### 41.1.10 wapt-get show-log

La commande `wapt-get show-log <nom du paquet>` renvoie les derniers journaux d’audit stockés dans la base de données sqlite locale de l’agent.

La commande prend un argument. Cet argument est le nom paquet ou une liste de noms de paquets avec le préfixe du référentiel.

La commande `wapt-get show tis-7zip` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Package: tis-7zip (21.06-34) PROD
-----
Status: OK

Installation log:
-----
Installing: 7z2106-x64.msi
Waiting for key key {23170F69-40C1-2702-2106-000001000000} to appear in Windows registry

Installation Parameters:
-----
{}

Last audit:
-----
Status: OK
Date: 2022-01-06T10:32:38.698272

Output:
Auditing tis-7zip
OK: Uninstall Key {23170F69-40C1-2702-2106-000001000000} in Windows Registry.

Next audit on: 2022-01-06T10:32:38.698272
```

### 41.1.11 wapt-get search

La commande `wapt-get search` permet de rechercher un ou plusieurs paquets dans les dépôts.

**Avertissement :** Cette commande renvoie uniquement les paquets disponibles pour l’hôte qui exécute la commande.  
Si une autre localisation, os, architecture ou maturité est présente dans le dépôt, elle n’est pas listée.

La commande de recherche prend un argument pour être recherchée dans le nom et la description du paquet.

La commande `wapt-get search "Firefox"` renvoie (par exemple) :

status	package	version	target_os	architecture	maturity	locale	description	repo
-	tis-firefox	94.0.2-106	windows	x64	PROD	fr	Mozilla Firefox est un	
	navigateur web gratuit et open source							
I	tis-config-firefox	68.3-6	windows	all	PROD		Configuration for Mozilla	
	Firefox - The package will not have any effect if an*							
I	tis-firefox-esr	91.3.0-105	windows	x64	PROD	fr	Mozilla Firefox Extended	
	Support Release (ESR) est une version officielle de*							

Value	status	package	version	target_os	architecture	maturity	locale	description	repo
Description	État de l'installation des paquets	Nom du paquet	Version du paquet	OS cible (si défini)	Architecture du CPU (si définie)	Maturité du paquet (si défini)	Locale du paquet (si défini)	Description du paquet	Dossier du paquet sur le serveur

**Note :** La valeur de *status* définit l'état de l'installation comme suit

- - pour non installé
- I pour installé

## 41.1.12 wapt-get download

La commande `wapt-get download <nom du paquet>` télécharge le paquet WAPT dans le cache local situé dans `C:\Program Files\wapt\cache`.

La commande prend un argument. Cet argument est le nom paquet ou une liste de noms de paquets avec le préfixe du référentiel.

La commande `wapt-get download tis-7zip` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Downloading packages tis-7zip(=19.00-25)
https://srvwapt.mydomain.lan/wapt/tis-7zip_19.00-25_x64_windows_0f4137ed1502b5045d6083aa258b5c42_5.
↳ 0_PROD_a10c57d7848cf7b145d6cd64bf4d5389.wapt : 1704227 / 1704227 (100%) (11804 KB/s)

Downloaded packages :
  C:\Program Files (x86)\wapt\cache\tis-7zip_19.00-25_x64_windows_0f4137ed1502b5045d6083aa258b5c42_5.0_PROD_a10c57d7848cf7b145d6cd64bf4d5389.wapt
```

### 41.1.13 wapt-get download-upgrade

La commande `wapt-get download-upgrade` télécharge les paquets à mettre à niveau dans le cache WAPT local `C:\Program Files (x86)\wapt\cache`.

La commande `wapt-get download-upgrade` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 18466658 /↵
↪54313787 (34%) (32089 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 36390179 /↵
↪54313787 (67%) (32693 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 52684289 /↵
↪54313787 (97%) (31564 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 54313787 /↵
↪54313787 (100%) (30747 KB/s)

=== downloaded packages ===
C:\Program Files (x86)\wapt\cache\B8D346E7-DDDB-0013-5A8A-425CF3B6199E.wapt
C:\Program Files (x86)\wapt\cache\tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt
```

### 41.1.14 wapt-get list

La commande `wapt-get list` liste les paquets WAPT qui sont installés sur l'ordinateur.

La commande `wapt-get list` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
package                version      install_status  install_date      description
↪                        package_uuid
-----
↪-----
tis-7zip                21.06-34   OK              2021-12-10T14:57 7-Zip is a free and open-
↪source file archiver with a high compression ratio 717a30cc-0d44-42d1-9538-0f2f298d8603
tis-firefox             94.0.1-106 OK           2021-12-10T14:58 Mozilla Firefox is a
↪free and open-source web browser                    5a91f54a-3e27-44cf-a2b6-6b84012aa3a2
```

package	version	install status	install_date	description-S tasks	package_uuid
Nom du paquet	Version du paquet	Installation status	Date et heure de l'installation	Description du paquet	UUID unique du paquet

### 41.1.15 wapt-get list-upgrade

La commande `wapt-get list-upgrade` liste les paquets WAPT qui doivent être mis à niveau sur l'hôte.

La commande `wapt-get list-upgrade` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini

=== upgrade packages ===
tis-notepadplusplus(=8.2-10)
```

### 41.1.16 wapt-get -S tasks

La commande `wapt-get -S tasks` vérifie si certaines tâches sont en cours d'exécution ou en attente dans la file d'attente.

La commande `wapt-get -S tasks` renvoie :

```
About to speak to waptservice...
Running task 14: Uninstall of tis-vlc (task #14), status:
```

## 41.2 Utilisation de lignes de commande spéciales avec WAPT

### 41.2.1 wapt-get clean

La commande `wapt-get clean` supprime les paquets du dossier `C:\Program Files (x86)\wapt\cache`.

La commande `wapt-get clean` est lancée après chaque mise à niveau pour économiser de l'espace disque.

La commande `wapt-get clean` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Removed files:
C:\Program Files (x86)\wapt\cache\tis-mumble_1.2.3-1_all.wapt
C:\Program Files (x86)\wapt\cache\tis-vlc_1.2.3-2_all.wapt
```

### 41.2.2 wapt-get upgradedb

La commande `wapt-get upgradedb` met à jour le schéma de la base de données WAPT locale si nécessaire.

La commande `wapt-get upgradedb` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
WARNING upgrade db aborted: current structure version 20210420 is newer or equal to requested.
↳structure version 20210420
No database upgrade required, current 20210420, required 20210420
```

### 41.2.3 wapt-get add-upgrade-shutdown - wapt-get remove-upgrade-shutdown

Ces 2 commandes modifient ce fichier : C:\Windows\System32\GroupPolicy\Machine\Scripts\scripts.ini.

- La commande `wapt-get add-upgrade-shutdown` ajoute un objet de stratégie de sécurité locale **waptexit**, permettant l'exécution de **waptexit** à l'arrêt du système.

La commande `wapt-get add-upgrade-shutdown` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
0
```

Le `scripts.ini` contient :

```
[Shutdown]
0CmdLine = C:\Program Files (x86)\wapt\waptexit.exe
0Parameters =
```

- La commande `wapt-get remove-upgrade-shutdown` supprime l'objet de politique de sécurité locale **waptexit**, désactivant l'exécution de **waptexit** pendant l'arrêt du système.

La commande `wapt-get add-upgrade-shutdown` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
0
```

Le `scripts.ini` contient :

```
[Shutdown]
```

### 41.2.4 wapt-get register

La commande `wapt-get register [<description>]` rapporte l'inventaire matériel et logiciel de l'ordinateur au serveur d'inventaire WAPT.

---

**Indication :** Vous pouvez passer une description comme argument à la commande `register`, cette description sera affichée dans la console WAPT dans la colonne *description*.

Vous pouvez profiter de WAPT pour améliorer votre gestion informatique en affectant un nom d'utilisateur ou un numéro de série d'ordinateur comme descriptions pour vos hôtes.

---

**Note :** Si l'hôte est déjà enregistré, la nouvelle exécution avec une description met à jour les informations enregistrées.

Il n'est pas nécessaire de définir une description pour enregistrer l'hôte avec la ligne de commande.

---

La commande `wapt-get register "John Doe PC"` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Registering host against server: https://srwwapt.mydomain.lan
Host correctly registered against server https://srwwapt.mydomain.lan.
```

### 41.2.5 wapt-get unregister

La commande `wapt-get unregister` supprime l'inventaire matériel et logiciel de l'ordinateur sur le serveur d'inventaire WAPT.

La commande `wapt-get unregister` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Unregistering host from server: https://srvwapt.mydomain.lan
Please get login for api/v3/hosts_delete:admin
Password:
Host correctly unregistered against server https://srvwapt.mydomain.lan.
```

### 41.2.6 wapt-get inventory

La commande `wapt-get inventory` affiche toutes les informations relatives à l'inventaire local au format JSON.

La commande `wapt-get inventory` renvoie (en partie) :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
{
  "host_info": {
    "description": "John Doe PC",
    "system_manufacturer": "Xen",
    "system_productname": "HVM domU",
    "computer_name": "Documentation",
    "computer_fqdn": "Documentation.srvwapt.mydomain.lan",
    "dnsdomain": "mydomain.lan",
    "workgroup_name": "Documentation",
    "domain_name": null,
    "domain_controller": null,
    "domain_controller_address": null,
    "domain_info_source": "history",
    "networking": [
      {
        "iface": "{085AB96368A-05A3B96-43EC-B773-0C0BB96794D9}",
        "mac": "a2:1d:6e:fc:8d:e6",
        "addr": [
          {
            "addr": "192.168.0.1",
            "netmask": "255.255.255.0",
            "broadcast": "192.168.0.255",
            "connected": true
          },
          {
            "addr": "fe80::2437:567f:79c8:f964",
            "netmask": "ffff:ffff:ffff:ffff::/64",
            "broadcast": "fe80::ffff:ffff:ffff:ffff%3",
            "connected": true
          }
        ]
      }
    ]
  }
}
```

(suite sur la page suivante)

```

],
"gateways": [
"192.168.0.254"
],
"dns_servers": [
"192.168.0.11"
],
"connected_ips": [
"192.168.0.1",
"fe80::2437:567f:79c8:f964"
],
"mac": [
"a2:fc:1d:6e:8d:e6"
],
...

```

## 41.2.7 wapt-get update-status

La commande `wapt-get update-status` renvoie le statut local au serveur d'inventaire WAPT.

**Note :** Si un composant matériel a été modifié sur l'ordinateur, le `update-status` ne renvoie pas cette information au serveur d'inventaire WAPT.

Pour ce faire, la commande à utiliser est `inventory`.

La commande `wapt-get update-status` renvoie :

```

Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Updated host status correctly sent to server https://srvwapt.mydomain.lan. {'success': True, 'msg':
→ 'update_host', 'result': {'uuid': 'B8D346E7-DDDB-0013-5A8A-425CF3B6199E', 'computer_fqdn':
→ 'documentation.mydomain.lan', 'status_hashes': {'dmi': '124b8bcef5b690afea7cf8001351a22132885123',
→ 'wmi': 'ae5dbb5627b7b3a5a31d5914a9dbf48b85b133da', 'host_info':
→ 'e737a82da15fbe9cae88ba9b4a9662a73657d959', 'audit_data': None, 'wapt_status':
→ 'bcb76ad07cf1b6f814082ec5a58c4fee0364a640', 'audit_status':
→ 'c34adb535c711b59d4408f00f77b7392687d7e56', 'host_metrics':
→ '9fc68bd98c82e0e9bece0ce3afaeeb63a3ed1db1', 'waptwua_status':
→ '4f9dcf0af339ce28d7354283fd4e6bdaf17b85c8', 'waptwua_updates':
→ 'c5cf38908fc549f499ade5b17ce221ff0ced377f', 'wuauserv_status':
→ '7c30215c3c34566e5b0c69c9e1dbfe3e6117b837', 'host_capabilities':
→ 'c31286122a213f3bb313531541582bb2ba1d0a81', 'installed_packages':
→ '3279f3bf4d5ed5086b198fa94a6a6f422f519ab3', 'last_update_status':
→ '347c5a8c01e182f1e03e5c9d0fe07dd87ab79153', 'installed_softwares':
→ 'd582a6f7325af35eae17cb7ecdca59ef0d137dda', 'authorized_certificates':
→ '2974f9535f813fc454b735193c31828b132a6ba0', 'waptwua_updates_localstatus':
→ 'c5cf38908fc549f499ade5b17ce221ff0ced377f'}, 'server_uuid': '82295c4d-4944-11ec-bac6-a25b5d7da3d5
→ '}, 'request_time': 0.046843767166137695}

```



### 41.2.8 wapt-get setlocalpassword

La commande `wapt-get setlocalpassword` permet de définir un mot de passe local pour les installations de paquets WAPT.

La commande `wapt-get setlocalpassword` renvoie :

```
Local password:
Confirm password:
Local auth password set successfully
```

### 41.2.9 wapt-get reset-uuid

La commande `wapt-get reset-uuid` récupère l'hôte *UUID* du BIOS et le renvoie au serveur d'inventaire WAPT.

La commande `wapt-get reset-uuid` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
New UUID: B0F23D44-86CB-CEFE-A8D6-FB8E3343FE7F
```

### 41.2.10 wapt-get generate-uuid

La commande `wapt-get generate-uuid` crée un nouvel hôte *UUID* et le renvoie au serveur d'inventaire WAPT.

À utiliser si vous avez un *bios UUID bug*.

La commande `wapt-get generate-uuid` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
New UUID: RND-0279A1F4-BBBE-43AE-A591-F82652E0104B
```

---

**Note :** Tous les UUID générés aléatoirement mettent un RND- devant.

---

### 41.2.11 wapt-get get-server-certificate

La commande `wapt-get get-server-certificate` télécharge le certificat SSL du serveur WAPT pour utiliser HTTPS pour communiquer avec le serveur WAPT.

Le certificat téléchargé est stocké dans `C:\Program Files(x86)\wapt\server\`.

La commande `wapt-get get-server-certificate` renvoie :

```
Server certificate written to C:\Program Files (x86)\wapt\ssl\server\srwapt.mydomain.lan.crt
```

### 41.2.12 wapt-get enable-check-certificate

La commande `wapt-get enable-check-certificate` télécharge le certificat SSL du serveur WAPT et active la communication sécurisée avec le serveur.

Il est utilisé pour *activer la vérification du certificat SSL / TLS*

La commande `wapt-get enable-check-certificate` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Server certificate : C:\Program Files (x86)\wapt\ssl\server\template-auto.test.lan.crt
Certificate CN: template-auto.test.lan
Pining certificate C:\Program Files (x86)\wapt\ssl\server\template-auto.test.lan.crt
```

### 41.2.13 wapt-get check-upgrades

La commande `wapt-get check-upgrades` montre l'état des dernières mises à jour/mises à niveau sur l'hôte.

La commande `wapt-get check-upgrades` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
{
  "running_tasks": [],
  "errors": [],
  "date": "2022-01-06T14:09:26.554391",
  "upgrades": [
    "test-notepadplusplus(=8.2-10)"
  ],
  "pending": {
    "install": [],
    "upgrade": [
      "test-notepadplusplus(=8.2-10)"
    ],
    "additional": [],
    "remove": [],
    "immediate_installs": []
  }
}
```

### 41.2.14 wapt-get add-licence

La commande `wapt-get add-licence` ajoute la licence sur le serveur.

La commande `wapt-get add-licence` renvoie :

```
Using config file C:\Program Files (x86)\wapt\wapt-get.ini
Server: https://srvwapt.mydomain.lan
Server UUID: 82295c4d-4944-11ec-bac6-a25b5d7da3d5
Server CABundle: 0

{"licence_nr":"6f011e23-cb70-40a4-b340-0d18ae1e2f02","product":"WAPT Enterprise","features":["full
```

(suite sur la page suivante)

(suite de la page précédente)

```

↪ "], "licenced_to": "documentation", "domain": "", "contact_email": "documentation@tranquil.it", "count":
↪ "10", "valid_from": "2021-06-14T00:00:00", "valid_until": "2022-01-12T00:00:00", "renewal_url": null,
↪ "signed_attributes": ["licence_nr", "product", "features", "licenced_to", "domain", "contact_email",
↪ "count", "valid_from", "valid_until", "renewal_url", "signed_attributes", "signer", "signature_date",
↪ "signer_certificate", "server_uuid"], "signer": "", "signature_date": "2022-01-13T16:38:56", "signer_
↪ certificate": "-----BEGIN CERTIFICATE-----\nMIIEIjCCAwwqAwIBAgIUUIOMdx8FmRdmCNTHxOfKecSp/
↪ cAAwDQYJKoZIhvcNAQEL\nBQAwZCcxZCZAJBgNVBAYTAkZSMStwIAYDVQQgHDB1TYWludCBTZWJhc3RpZW4gc3Vy\
↪ nIEvvaXJlMRwwGgYDVQQKDBUcmFucXVpbCBJVCBTeXN0ZW1zMSAwHgYDVQQDDDBdy\
↪ nZWxpY2VuY2luZy50cmFucXVpbC5pdDEkMCIgCSqGSIsb3DQeJARYVdGVjaG5pcXV1\
↪ nQHRyYW5xdWlsLml0MB4XDTIwMDYwODE0MTQ0MVoXDTMxMDYwNjE0MTQ0MVoWZCcx\
↪ nCzAJBgNVBAYPkdZrIrw9Kb5hee+1EgqEbudCBTZWJhc3RpZW4gc3VyIEvvaXJl\
↪ nMRwwGgYDVQQKDBUcmFucXVpbCBJVCBTeXN0ZW1zMSAwHgYDVQQDDDBdyZWxpY2Vu\
↪ nY2luZy50cmFucXVpbC5pdDEkMCIgCSqGSIsb3DQeJARYVdGVjaG5pcXV1QHRyYW5x\
↪ ndWlsLml0MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAz43W80hWXAE\nhDB+IWwQm9IGGdR0VY/k1KcSheo/
↪ 8jG1NziyH6BANhjFKYXN9UtQ+ghzv6BGfSTH\nnyua1aXEQM89sSKF0oJzt0D1L9FZtuWQb/vfLWkisP8fRPvH4B/
↪ tYG+5nchGa6+6r\nqGSGSpWcnj6CovgQR01ATUuHN3NV1N7q48hBT/ZT9R5U3sEi+hNK4eRIeZ220Pzm\
↪ nDoNgkVK1EiczgXuM77ezYp8UwvPk6dZrIrw9Kb5hee+1EgqEbgVmdARoaOPGTK8h\
↪ n8VW+milWsl4TEY19kxXWvva+M6wX00ipJ2LxEiu5+d10ok9E8i405UTNE7oSVYsF\n90/
↪ 6S3C4twIDAQABo2QwYjAPBgNVHRMBAf8EBTADAQH/MCAGA1UdJQEB/wQWMBQG\
↪ nCCsGAQUFBwMDBggrBgEFBQcDAjAdBgNVHQ4EFgQUpRT6Co2uoWZMCwP7FKiF73+j\nfAEwDgYDVR0PAQH/
↪ BAQDAgHwMA0GCSqGSIsb3DQeBCwUAA4IBAQAAdXX5IkpuH/Gek\nPPHC4KvE/
↪ 6GsU0kqLI1w5ML5pbF1zyCCL0nm4f8w2JJJ2YcDb4QVD27kJqgZch1\nnniYQ3RCIh6aasS8qpCof90KkpVKMjiyk/
↪ ra7I6NSgPut4ErkoxUWocgF6SNFEjwB\nnaqUZY//Hkoqk2dXqdujLVGJfBpX95ZJ9PmFNLfsyUsvu1WcFMb0En0EU074Mq4M3\
↪ nKo2S86G9pEDKooaNS5Vq19biReOwQYpX1Y1SLtrxFx8AM87auQgaD8EWSdA1q2ycN\
↪ n8ZnmXGxAhDv8hmE2Fv0x0t3hzYXxcv1ZjYWRH1MUL/buWQQ35u9MFkj7YZ1T1M\nnb9wjtN+W\n-----END CERTIFICATE-
↪ -----\n", "signature":
↪ "J7DZ+mja7zGghYFCDKh8WIXzzdhKPeoNswWjnKZziT+ddpoRdg45kZz4E8PxMIUzhTI8WIXzzdhKPeoNswrICpQ8t5kepzovZpo0Nwjg00
↪ de18bEgSSlgjXgE/wr2ZfclsRsRRfsRbGsterRKQcthNDRflf8RjH5cpDnDvMJ+qJtTsqaA13/WT2NS2uNWZI93si/
↪ 9mowWY8MdT/PZjoscicqjibq4oa+/FrPsAlHU0tcGE9JylwknzsUD5Ayfh+9sNLLxsG6eT0JlnNgf4nx9mXAU4GBg==",
↪ "server_uuid": "82295c4d-4944-11ec-bac6-a25b5d7da3d5"}
Login to server api/v3/licences
Waptserver https://srvwapt.mydomain.lan Admin User () :admin
Waptserver Password: *****
Licence properly activated on server

```

### 41.2.15 wapt-get check-licences

La commande `wapt-get check-licences` affiche les licences enregistrées sur le serveur.

La commande `wapt-get check-licences` renvoie :

```

Using config file C:\Program Files (x86)\wapt\wapt-get.ini
Server: https://srvwapt.mydomain.lan
Server UUID: 36bf01bc-c8f5-11eb-bf04-36127be97253
Server CABundle: 0

Total licences count: 10
Licenced to: documentation

```

(suite sur la page suivante)

(suite de la page précédente)

```
Valid Nr:b7b6e537-3cb7-4d9a-3cb7-2448020e2e51 Count:10 From:2022-01-13T00:00:00 Expire:2023-01-
↪12T23:59:00 Server:36bf01bc-c8f5-11eb-bf04-36127be97253 Licencee:documentation
```

### 41.2.16 wapt-get dnsdebug

La commande `wapt-get dnsdebug` montre les informations dns locales.

La commande `wapt-get dnsdebug` renvoie :

```
DNS Server : dns.mydomain.lan
DNS Domain : mydomain.lan
Main repo url: https://srvwapt.mydomain.lan/wapt
wapt SRV: []
waptserver SRV: []
CNAME: []
```

## 41.3 Utilisation de la ligne de commande pour la configuration de la session utilisateur

### 41.3.1 wapt-get session-setup

La commande `wapt-get session-setup <nom du paquet> [<ALL>]` lance les personnalisations de niveau utilisateur des paquets WAPT installés.

Elle est définie dans la fonction `session-setup` du fichier `setup.py`.

La commande prend un argument. Cet argument est le nom paquet ou une liste de noms de paquets avec le préfixe du référentiel.

---

**Note :** L'argument `ALL` lancera `session-setup` pour tous les paquets WAPT installés.

---

La commande `wapt-get session-setup ALL` renvoie :

```
Configuring tis-7zip ... No session-setup. Done
Configuring tis-ccleaner ... Already installed. Done
Configuring tis-vlc ... No session-setup. Done
Configuring mdl-tightvnc ... No session-setup. Done
Configuring tis-brackets ... No session-setup. Done
Configuring mdl-firefox-esr ... No session-setup. Done
Configuring tis-paint.net ... No session-setup. Done
```

## 41.4 Utilisation de la ligne de commande pour créer des paquets WAPT

### 41.4.1 wapt-get list-registry

La commande `wapt-get list-registry` renvoie une liste des logiciels installés présents dans le registre.

La commande peut prendre un argument, non sensible à la casse, pour rechercher un mot spécifique : `wapt-get list-registry <keyword>`.

Les informations retournées sont :

Information	Définition	Disponible sur Windows	Disponible sur Linux	Disponible sur Mac OS
<i>Uninstall-Key</i>	Désinstaller l'identifiant de clé dans le registre	✓	✗	✗
<i>Software</i>	Nom du logiciel dans le registre	✓	✓	✓
<i>Version</i>	Version du logiciel dans le registre	✓	✓	✓
<i>Uninstall-string</i>	Chaîne de désinstallation du logiciel dans le registre	✓	✗	✗

#### Note :

- Sous Windows, ces informations sont collectées par WAPT à partir du registre en deux localisations :
  - ComputerHKEY\_LOCAL\_MACHINESOFTWAREMicrosoftWindowsCurrentVersionUninstall
  - ComputerHKEY\_LOCAL\_MACHINESOFTWAREWOW6432NodeMicrosoftWindowsCurrentVersionUninstall
- Sous Linux, ces informations sont collectées par WAPT dans Applications
- Sous Mac OS, ces informations sont collectées par WAPT dans `/var/lib/dpkg/info/`

La commande `wapt-get list-registry firefox` renvoie (sous Windows) :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
UninstallKey          Software          Version
↳ Uninstallstring
-----
↳ -----
Mozilla Firefox 45.5.0 ESR (x64 fr)  Mozilla Firefox 45.5.0 ESR (x64 fr)  45.5.0
↳ "C:\Program Files\Mozilla Firefox\uninstall\helper.exe"
```

### 41.4.2 wapt-get sources

La commande `wapt-get sources <nom du paquet>` télécharge les sources depuis une plateforme de gestion du code source comme Git ou SVN.

La commande prend un argument. Cet argument est le nom paquet ou une liste de noms de paquets avec le préfixe du référentiel.

La commande `wapt-get sources tis-firefox` ne renvoie rien ;

### 41.4.3 wapt-get make-template

**Avvertissement :** Cette méthode ne doit être utilisée que si vous ne pouvez pas utiliser la console pour créer un paquet.

La commande `wapt-get make-template <chemin-installateur> [<nompaquet> [<nom-dossier source>]]` permet de créer un modèle de paquet à partir d'un binaire.

Utilisez ces arguments :

Arguments	Définition	Valeur par défaut
chemin de l'installateur	Chemin d'installation du binaire	Pas de valeur par défaut, c'est nécessaire
nompaquet	Nom du paquet	S'il n'est pas défini, le nom du paquet est créé comme ceci <code>prefix-binaire-nom_paquet-version-wapt</code>
nom du répertoire source	Répertoire pour enregistrer les fichiers sources	Si elle n'est pas définie, c'est la valeur de <code>default_sources_root</code> dans <code>waptconsole.ini</code>

La commande `C:\Users\User\Downloads\tightvnc.msi tis-tightvnc`` renvoie :

```
Using config file: C:\Users\Documentation\AppData\Local\waptconsole\waptconsole.ini
Template created. You can build the WAPT package by launching
C:\Program Files (x86)\wapt\wapt-get.exe build-package C:\waptdev\tis-tightvnc-wapt
You can build and upload the WAPT package by launching
C:\Program Files (x86)\wapt\wapt-get.exe build-upload C:\waptdev\tis-tightvnc-wapt
```

#### Indication :

- Si vous avez préalablement installé le paquet `tis-waptdev` sur votre ordinateur de développement, l'éditeur **PyScripter** se lancera automatiquement et ouvrira le paquet en mode développement.

### 41.4.4 wapt-get make-host-template

**Avvertissement :** Cette méthode ne doit être utilisée que si vous ne pouvez pas utiliser la console pour créer un paquet.

La commande `wapt-get make-host-template <nommachine> [[<paquet>,<paquet>,...] [répertoire]]` crée un paquetage hôte WAPT vide à partir d'un modèle.

Utilisez ces arguments :

Arguments	Définition	Valeur par défaut
nommachine	Nom de la machine utilisée pour le nom du paquet	Si aucun n'est donné, utiliser le FQDN
package	Liste des paquets nécessaires sur l'hôte.	S'il n'est pas défini, aucun paquet n'est ajouté comme dépendance
répertoire	Répertoire pour enregistrer les fichiers sources	S'il n'est pas défini, c'est <code>C:\waptdev</code>

La commande `wapt-get make-host-template host01.mydomain.lan` renvoie :

```
Using config file: C:\Users\Documentation\AppData\Local\waptconsole\waptconsole.ini
Template created. You can build the WAPT package by launching
  C:\Program Files (x86)\wapt\wapt-get.exe build-package C:\waptdev\host01.mydomain.lan-wapt
You can build and upload the WAPT package by launching
  C:\Program Files (x86)\wapt\wapt-get.exe build-upload C:\waptdev\host01.mydomain.lan-wapt
```

#### 41.4.5 wapt-get make-group-template

**Avertissement :** Cette méthode ne doit être utilisée que si vous ne pouvez pas utiliser la console pour créer un paquet.

La commande `wapt-get make-group-template <nom du groupe>` crée un paquet de groupe WAPT vide à partir d'un modèle.

La commande `wapt-get make-group-template documentation` renvoie :

```
Using config file: C:\Users\documentation\AppData\Local\waptconsole\waptconsole.ini
Template created. You can build the WAPT package by launching
  C:\Program Files (x86)\wapt\wapt-get.exe build-package c:\waptdev\documentation_0-wapt
You can build and upload the WAPT package by launching
  C:\Program Files (x86)\wapt\wapt-get.exe build-upload c:\waptdev\documentation_0-wapt
```

#### 41.4.6 wapt-get build-package

La commande `wapt-get build-package <chemin vers le répertoire package>` construit un package WAPT et le signe avec la clé privée associée au `personal_certificate_path` défini dans le `waptconsole.ini`.

La commande `wapt-get build-package c:\waptdev\tis-dropbox` renvoie :

```
Using config file: C:\Users\documentation\AppData\Local\waptconsole\waptconsole.ini
Building packages 1 packages
Personal certificate is documentation
Please type the password to decrypt the private key C:\Users\documentation\private\documentation.pem
Password:
Private key is <SSLPrivateKey 'C:\\Users\\documentation\\private\\documentation.pem'>
Personal certificate is SSLCertificate cn=documentation
Private key is <SSLPrivateKey 'C:\\Users\\documentation\\private\\documentation.pem'>
Building  c:\waptdev\tis-dropbox
Signing c:\waptdev\tis-dropbox with key <SSLPrivateKey 'C:\\Users\\documentation\\private\\
↪documentation.pem'> and certificate documentation (C:\Users\documentation\private\documentation.
↪crt)
Package c:\waptdev\tis-dropbox signed : signature : BN7j6lwloY...Iu9QVula=
...done building. Package filename c:\waptdev\tis-dropbox_104.4.175-7_windows_
↪0f4137ed1502b5045d6083aa258b5c42_5.1_PROD.wapt
1 packages successfully built
0 packages failed

You can upload to repository with
```

(suite sur la page suivante)

(suite de la page précédente)

```
C:\Program Files (x86)\wapt\wapt-get.exe upload-package "c:\waptdev\tis-dropbox_104.4.175-7_
↪windows_0f4137ed1502b5045d6083aa258b5c42_5.1_PROD.wapt"
```

**Avertissement :** Le nom du répertoire ne définit pas le nom et le préfixe du paquet, il est défini par le fichier `control`.

### 41.4.7 wapt-get sign-package

La commande `wapt-get sign-package <chemin vers le paquet>` signe un paquet avec la clé privée associée à `personal_certificate_path` définie dans le `waptconsole.ini`.

**Attention :** `sign-package` ne renomme pas le paquet WAPT avec le préfixe choisi de l'*Organisation*.

La commande `wapt-get sign-package c:\waptdev\tis-dropbox` renvoie :

```
Using config file: C:\Users\documentation\AppData\Local\waptconsole\waptconsole.ini
Signing packages c:\waptdev\tis-dropbox
Personal certificate is SSLCertificate cn=documentation
Please type the password to decrypt the private key C:\Users\documentation\private\documentation.pem
Password:
Private key is <SSLPrivateKey 'C:\\Users\\documentation\\private\\documentation.pem'>
Signing c:\waptdev\tis-dropbox
OK: Package c:\waptdev\tis-dropbox signed : signature : b'nJYfYswDWi'...b'v790D7uA='
```

### 41.4.8 wapt-get build-upload

La commande `wapt-get build-upload <chemin vers le paquet>` construit et télécharge un paquet WAPT sur le dépôt WAPT principal.

**Indication :** En passant l'argument `-i` à `build-upload`, le numéro de version du packaging WAPT est incrémenté avant le téléchargement, afin d'éviter de devoir modifier manuellement le fichier `control`.

La commande `wapt-get -i build-upload c:\waptdev\tis-dropbox` renvoie :

```
Using config file: C:\Users\documentation\AppData\Local\waptconsole\waptconsole.ini
Building packages 1 packages
Personal certificate is documentation
Please type the password to decrypt the private key C:\Users\documentation\private\documentation.pem
Password:
Private key is <SSLPrivateKey 'C:\\Users\\documentation\\private\\documentation.pem'>
Personal certificate is SSLCertificate cn=documentation
Private key is <SSLPrivateKey 'C:\\Users\\documentation\\private\\documentation.pem'>
Building c:\waptdev\tis-dropbox
Signing c:\waptdev\tis-dropbox with key <SSLPrivateKey 'C:\\Users\\documentation\\private\\
↪documentation.pem'> and certificate documentation (C:\Users\documentation\private\documentation.
```

(suite sur la page suivante)



(suite de la page précédente)

```

↪ crt)
Package c:\waptdev\tis-dropbox signed : signature : s9FOLFQvYw...c9T3Hv1A=
...done building. Package filename c:\waptdev\tis-dropbox_104.4.175-7_windows_
↪ 0f4137ed1502b5045d6083aa258b5c42_5.1_PROD.wapt
1 packages successfully built
0 packages failed
Building and uploading packages to https://srvwapt.mydomain.lan
Please get login for https://srvwapt.mydomain.lan/api/v3/upload_xxx:admin
Password:
c:\waptdev\tis-dropbox_104.4.175-7_windows_0f4137ed1502b5045d6083aa258b5c42_5.1_PROD.
↪ wapt[=====] 126459984/126459984 - 00:00:40
Package uploaded successfully: 1 Packages uploaded, 0 errors

```

### 41.4.9 wapt-get duplicate

La commande `wapt-get duplicate <répertoire ou paquet> <nouveau-nom-paquet> [<nouvelle-version> [<répertoire cible>]]` duplique un paquet.

**Avertissement :** Ne l'utilisez pas pour le paquet hôte.

Utilisez ces arguments :

Arguments	Définition	Requis
dossier ou paquet	Chemin du répertoire du paquet ou du paquet compilé dans <code>.wapt</code>	✓
nouveau-nom-paquet	Nom du nouveau paquet	✓
nouvelle-version	Change la version du paquet dans le fichier control. S'il n'est pas défini, la même version est dupliquée.	✗
répertoire cible	Chemin vers le répertoire cible du paquet dupliqué. S'il n'est pas défini, utilise le même que celui du paquet source.	✗

La commande `wapt-get duplicate tis-firefox tis-firefox-custom` renvoie :

```

Using config file: C:\Users\documentation\AppData\Local\waptconsole\waptconsole.ini
Package duplicated. You can build the new WAPT package by launching
  C:\Program Files (x86)\wapt\wapt-get.exe build-package C:\waptdev\tis-firefox-custom-wapt
You can build and upload the new WAPT package by launching
  C:\Program Files (x86)\wapt\wapt-get.exe build-upload C:\waptdev\tis-firefox-custom-wapt

```

**Indication :**

- Si vous avez préalablement installé le paquet `tis-waptdev` sur votre ordinateur de développement, l'éditeur **PyScripter** se lancera automatiquement et ouvrira le paquet en mode développement.

## 41.4.10 wapt-get edit

**Avertissement :** Cette méthode ne doit être utilisée que si vous ne pouvez pas utiliser la console pour créer un paquet.

La commande `wapt-get edit <nom du paquet>` télécharge et édite un paquet WAPT depuis le dépôt du serveur.

La commande prend un argument. Cet argument est le nom paquet ou une liste de noms de paquets avec le préfixe du référentiel.

La commande `wapt-get edit tis-firefox` renvoie :

```
Using config file: C:\Users\documentation\AppData\Local\waptconsole\waptconsole.ini
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 1629411 /_
↪54313787 (3%) (2686 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 8147055 /_
↪54313787 (15%) (5679 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 15207836 /_
↪54313787 (28%) (7367 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 19552932 /_
↪54313787 (36%) (7249 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 24984302 /_
↪54313787 (46%) (7471 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 29329398 /_
↪54313787 (54%) (7143 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 33674494 /_
↪54313787 (62%) (6951 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 40735275 /_
↪54313787 (75%) (7534 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 45623508 /_
↪54313787 (84%) (7326 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 53227426 /_
↪54313787 (98%) (7603 KB/s)
https://srvwapt.mydomain.lan/wapt/tis-firefox_94.0.1-106_x64_windows_
↪0f4137ed1502b5045d6083aa258b5c42_6.1_PROD_en_f5335369ad5805e8dbc1f8ae99b2258e.wapt : 54313787 /_
↪54313787 (100%) (7663 KB/s)
Package edited. You can build and upload the new WAPT package by launching
```

### Indication :

- Si vous avez préalablement installé le paquet `tis-waptdev` sur votre ordinateur de développement, l'éditeur **PyScripter** se lancera automatiquement et ouvrira le paquet en mode développement.

### 41.4.11 wapt-get edit-host

**Avertissement :** Cette méthode ne doit être utilisée que si vous ne pouvez pas utiliser la console pour créer un paquet.

La commande `wapt-get edit-host <host FQDN>` édite un paquet WAPT *host*.

La commande `wapt-get edit-host RND-0279A1F4-BBBE-43AE-A591-F82652E0104B` renvoie :

```
Using config file: C:\Users\documentation\AppData\Local\waptconsole\waptconsole.ini
Package edited. You can build and upload the new WAPT package by launching
C:\Program Files (x86)\wapt\wapt-get.exe -i build-upload c:\waptdev\RND-0279A1F4-BBBE-43AE-A591-
↪F82652E0104B_0-wapt
```

#### Indication :

- Si vous avez préalablement installé le paquet `tis-waptdev` sur votre ordinateur de développement, l'éditeur **PyScripter** se lancera automatiquement et ouvrira le paquet en mode développement.

### 41.4.12 wapt-get update-package-sources

La commande `wapt-get update-package-sources <chemin vers le paquet>` exécute la fonction `update-package` dans le fichier `setup.py`.

La commande `wapt-get update-package-sources tis-firefox` renvoie :

```
Using config file: C:\Users\documentation\AppData\Local\waptconsole\waptconsole.ini
Latis Mozilla Firefox version is: 95.0.2
Download URL is: https://download-installer.cdn.mozilla.net/pub/firefox/releases/95.0.2/win64/en-US/
↪Firefox%20Setup%2095.0.2.exe
Downloading: Firefox_Setup_95.0.2.exe
Firefox_Setup_95.0.2.exe[=====] 54810424/54810424 - 00:00:07
Software version updated (from: 94.0.1 to: 95.0.2)
Packages updated :
c:\waptdev\tis-firefox_0-wapt
```

#### Indication :

- Si vous avez préalablement installé le paquet `tis-waptdev` sur votre ordinateur de développement, l'éditeur **PyScripter** se lancera automatiquement et ouvrira le paquet en mode développement.

## 41.5 Utilisation des lignes de commande pour la gestion de WaptWUA

### 41.5.1 wapt-get waptwua-scan

Le `wapt-get waptwua-scan` analyse l'état de Windows par rapport aux règles actuelles et envoie le résultat au serveur.

La commande `wapt-get waptwua-scan` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Ensure wuauerv Auto Update option is disabled
Scanning with windows updates rules:
{
"direct_download": false,
"default_allow": false,
"filter": "Type='Software' or Type='Driver'",
"download_scheduling": "7d",
"install_scheduling": null,
"install_delay": null,
"postboot_delay": "10m"
}
Downloading wsusscn2.cab file from https://srvwapt.mydomain.lan/waptwua/wsusscn2.cab
wsusscn2.cab[=====] 1024297844/1024297844 - 00:02:04
Windows updates rules have been changed
Looking for updates with filter: Type='Software' or Type='Driver'
  Connecting to local update searcher using offline wsusscn2 file...
  Offline Update searcher ready...
Waiting for WUA search to complete
Done searching
WUA Search completed !
Updates scan done.
Writing status in local wapt DB
Status: OK
(0, 0, 0)
None
re-enabling wuauerv previous state: 0
```

### 41.5.2 wapt-get waptwua-download

La commande `wapt-get waptwua-download` analyse l'état de Windows par rapport aux règles en vigueur, télécharge les kb manquantes et envoie le résultat au serveur.

La commande `wapt-get waptwua-download` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Ensure wuauerv Auto Update option is disabled
Downloading wsusscn2.cab file from https://srvwapt.mydomain.lan/waptwua/wsusscn2.cab
wsusscn2.cab[=====] 1024297844/1024297844 - 00:00:26
Start of install for all pending Windows updates
Scanning with params:
{
```

(suite sur la page suivante)

(suite de la page précédente)

```

"direct_download": false,
"default_allow": false,
"filter": "Type='Software' or Type='Driver'",
"download_scheduling": "7d",
"install_scheduling": null,
"install_delay": null,
"postboot_delay": "10m"
}
Scanning with windows updates rules:
{
"direct_download": false,
"default_allow": false,
"filter": "Type='Software' or Type='Driver'",
"download_scheduling": null,
"install_scheduling": null,
"install_delay": null,
"postboot_delay": "10m"
}
Bypassing scan, no change since last successful scan
Writing status in local wapt DB
Status: OK
{'downloaded': [], 'missing': []}
None
re-enabling wuauerv previous state: 0

```

### 41.5.3 wapt-get waptwua-install

Le `wapt-get waptwua-install` installe les mises à jour en attente.

La commande `wapt-get waptwua-install` renvoie :

```

Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Ensure wuauerv Auto Update option is disabled
Downloading wsusscn2.cab file from https://srvwapt.mydomain.lan/waptwua/wsusscn2.cab
wsusscn2.cab[=====] 1024297844/1024297844 - 00:00:26
Start of install for all pending Windows updates
Scanning with params:
{
"direct_download": false,
"default_allow": false,
"filter": "Type='Software' or Type='Driver'",
"download_scheduling": "7d",
"install_scheduling": null,
"install_delay": null,
"postboot_delay": "10m"
}
Scanning with windows updates rules:
{
"allowed_products": null,

```

(suite sur la page suivante)

```
"allowed_classifications": null,
"allowed_severities": null,
"allowed_updates": null,
"forbidden_updates": null,
"allowed_kbs": null,
"forbidden_kbs": null,
"default_allow": false
}
Looking for updates with filter: Type='Software' or Type='Driver'
  Connecting to local update searcher using offline wsusscn2 file...
  Offline Update searcher ready...
Waiting for WUA search to complete
Done searching
WUA Search completed !
Updates scan done.
Installed 07609d43-d518-4e77-856e-d1b316d1b8a8 : MSXML 6.0 RTM Security Update (925673)
Installed bb49cc19-8847-4986-aa93-5e905421e55a : Security Update for Microsoft Visual C++ 2005_
↳Service Pack 1 Redistributable Package (KB2538242)
Installed 729a0dcb-df9e-4d02-b603-ed1aee074428 : Security Update for Microsoft Visual C++ 2008_
↳Service Pack 1 Redistributable Package (KB2538243)
Installed 719584bc-2208-4bc9-a650-d3d6347eb32e : Security Update for Microsoft Visual C++ 2010_
↳Service Pack 1 Redistributable Package (KB2565063)
Installed a8761130-35b6-41ce-8b67-2d35bb2d0846 : 2021-02 Cumulative Update for .NET Framework 3.5_
↳and 4.8 for Windows 10, version 20H2 for x64 (KB4601050)
Installed 30f75e5d-2c46-42be-aef6-97ae730452be : 2021-07 Cumulative Update for Windows 10 Version_
↳20H2 for x64-based Systems (KB5004945)
Installed 6e88be6e-d470-4e7e-9f36-01479049aadb : 2021-08 Servicing Stack Update for Windows 10_
↳Version 20H2 for x64-based Systems (KB5005260)
Installed a15155a4-1299-41ff-9a39-28a33ce7cadd : 2021-12 .NET Core 3.1.22 Security Update for x64_
↳Client (KB5009193)
Installed 38db0ad6-27f8-4bf9-ab2a-cffc4d7bc390 : Windows Malicious Software Removal Tool x64 - v5.
↳96 (KB890830)
Scanning with windows updates rules:
{
"direct_download": false,
"default_allow": false,
"filter": "Type='Software' or Type='Driver'",
"download_scheduling": "7d",
"install_scheduling": null,
"install_delay": null,
"postboot_delay": "10m"
}
Windows updates rules have been changed
Writing status in local wapt DB
Status: OK
[]
None
re-enabling wuauerv previous state: 2
```

## 41.5.4 waptwua-status

Le `wapt-get waptwua-status` renvoie le statut de la dernière mise à jour.

La commande `wapt-get waptwua-status` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
{'enabled': None,
 'last_error': 'OperationalError: cannot rollback - no transaction is active',
 'last_install_batch': [],
 'last_install_date': None,
 'last_install_reboot_required': None,
 'last_install_result': None,
 'last_scan_date': '2022-01-07T10:20:50.213644',
 'last_scan_duration': 1490.500022649765,
 'missing_downloads': [],
 'rules_packages': [],
 'status': 'SCANNING',
 'wsusscn2cab_date': '2021-12-14T04:06:46'}
None
```

## 41.6 Utilisation des lignes de commande pour l'interaction avec les utilisateurs

### 41.6.1 wapt-get propose-upgrade

La commande `wapt-get propose-upgrade` lance une proposition de mise à niveau en lançant `waptexit` dans les sessions ouvertes.

La commande `wapt-get propose-upgrade` renvoie :



```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
{'result': 1, 'summary': 'waptexit launched for 1 sessions'}
```

## 41.7 Utilisation des lignes de commande pour la configuration initiale

### 41.7.1 wapt-get create-keycert

La commande `wapt-get create-keycert [<options>]` crée une paire de clés RSA et un certificat X509.

TABLEAU 1 – Liste des options disponibles

Option	Description	Valeur par défaut
--CommonName	Nom d'affichage du certificat	/
--CommonName64	Nom d'affichage du certificat, encodé en base64 (si accents, espaces etc..)	/
--CodeSigning 	Si le certificat / la paire de clés sera autorisé(e) à signer des paquets logiciels	0
--CA 	Si ce certificat / paire de clés peut être utilisé pour signer d'autres certificats (Autorité de certification principale ou intermédiaire)	0
--ClientAuth	Correspond à une propriété (utilisation) du certificat	1 pour un certificat non-CA
--PrivateKeyPassword	Mot de passe pour déverrouiller la clé si --NoPrivateKeyPassword n'est pas utilisé	Mot de passe généré aléatoirement
--PrivateKeyPassword64 si --PrivateKeyPassword n'est pas utilisé	Mot de passe pour déverrouiller la clé, codé en base64 (si accents, espaces etc.)	Mot de passe généré aléatoirement
--NoPrivateKeyPassword	Si la clé privée n'est pas chiffrée si --PrivateKeyPassword ou --PrivateKeyPassword64 ne sont pas utilisés	Vide
-F	Forcer l'écrasement du certificat existant	/
--Country	Nom du pays du titulaire du certificat à enregistrer dans le certificat.	/
--Locality	Nom de la ville du titulaire du certificat à inscrire dans le certificat.	/
--Organization	Nom de l'organisation du titulaire du certificat à enregistrer dans le certificat.	/
--OrgUnit	Nom de l'unité d'organisation (service) du titulaire du certificat à enregistrer dans le certificat.	/
--Email	Adresse e-mail du détenteur du certificat à enregistrer dans le certificat	/
--CAKeyFilename	Chemin vers la clé (.pem) d'une Autorité de Certification	Paramètre default_ca_key_path dans waptconsole.ini
--CACertFilename	Chemin vers le certificat (.crt) d'une Autorité de Certification	Paramètre default_ca_cert_path dans waptconsole.ini
--CAKeyPassword	Mot de passe pour déverrouiller la clé de l'Autorité de Certification	/
-NoCAKeyPassword	Si la clé de l'autorité de certification n'est pas chiffrée	/
--BaseDir	Dossier où la clé privée et le certificat public seront stockés.	Répertoire de personal_certificate_path dans waptconsole.ini
-EnrollNewCert	Copie le certificat dans waptssl	/
-SetAsDefaultPersonalCert	Le chemin du certificat est assigné à personal_certificate_path dans waptconsole.ini	/

La commande `wapt-get create-keycert` renvoie :



```
Using config file C:\Users\Administrator\AppData\Local\waptconsole\waptconsole.ini
BaseDir: C:\private\
Common name of certificate to create: documentation
Private Key Filename: C:\private\documentation.pem
Certificate Filename: C:\private\documentation.crt
New private key password: QR.-DVp6MPGW
```

**Avertissement** : Si `default_ca_key_path` et `default_ca_cert_path` sont définis dans `C:\Users\Administrator\AppData\Local\waptconsole\waptconsole.ini`, vous devez placer le certificat CA au même endroit.

Sinon cette erreur apparait :

```
wapt-get create-keycert
Using config file C:\Users\tisadmin\AppData\Local\waptconsole\waptconsole.ini
BaseDir: C:\Users\tisadmin\private\
Common name of certificate to create: CRT
Exception at 00483595: Exception:
CA Certificate C:\Program Files (x86)\wapt\ssl does not exist.
```

## 41.7.2 wapt-get build-waptagent

Le `wapt-get build-waptagent` [`</ConfigFilename>`] compile et télécharge un paquet **waptagent.exe** et **waptupgrade.exe** en utilisant le paramètre `/ConfigFilename` du fichier ini. Pour le fichier ini, utilisez la syntaxe `wapt-get.ini`.

**Note** : Par défaut, utilise la configuration `waptconsole.ini`.

La commande `wapt-get build-waptagent` renvoie :

```
Building customized waptagent.exe installer
.....
Built C:\Users\documentation\AppData\Local\Temp\wapt20220107T122037000000.tmp\waptupgrade\waptagent.
↪exe
Private key Password for C:\Users\documentation\private\documentation.crt : *****
Building waptupgrade package
Waptserver https://srvwapt.mydomain.lan Admin User () :admin
Waptserver Password: *****
Uploading customized waptagent.exe installer
Uploading C:\Users\documentation\AppData\Local\Temp\wapt20220107T122037000000.tmp\waptupgrade\
↪waptagent.exe to waptserver https://srvwapt.mydomain.lan
OK
Uploading C:\Users\documentation\AppData\Local\Temp\wapt20220107T122037000000.tmp\test-waptupgrade_2.
↪1.2.10605-0_all_PROD_all.wapt to waptserver https://srvwapt.mydomain.lan
OK : 1 Packages uploaded, 0 errors. Errors:
```

## 41.8 Utilisation des lignes de commande pour la gestion des dépôts

### 41.8.1 wapt-get upload-package

La commande `wapt-get upload-package <chemin vers le paquet>` télécharge un paquet sur le dépôt principal de WAPT.

La commande `wapt-get upload-package C:waptdev is-tightvnc.wapt` renvoie :

```
Using config file: C:\Users\documentation\AppData\Local\waptconsole\waptconsole.ini
Uploading packages to https://srvwapt.mydomain.lan
Please get login for https://srvwapt.mydomain.lan/api/v3/upload_xxx:admin
Password:
c:\waptdev\tis-tightvnc.wapt[=====] 54316019/54316019 - 00:00:17
OK : 1 Packages uploaded, 0 errors
```

### 41.8.2 wapt-get scan-packages

---

**Indication :** Cette commande ne concerne que les dépôts Windows

---

La commande `wapt-get scan-packages <dossier>` reconstruit un fichier Packages pour le dépôt de paquets http.

La commande `wapt-get scan-packages C:waptwaptserverrepositorywapt` renvoie :

```
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Packages filename: C:\wapt\waptserver\repository\wapt
Processed packages:
  C:\wapt\waptserver\repository\wapt\tis-firefox.wapt
  C:\wapt\waptserver\repository\wapt\tis-tightvnc.wapt
  C:\wapt\waptserver\repository\wapt\tis-7zip.wapt
Skipped packages:
```

### 41.8.3 wapt-scanpackages

---

**Indication :** Cette commande est seulement pour les dépôts Linux

---

La commande `wapt-scanpackages <dossier>` reconstruit un fichier Packages pour le dépôt de paquets http.

La commande `wapt-scanpackages /var/www/wapt/` ne renvoie rien

## 41.8.4 Re-signature des paquets sur le serveur à l'aide d'une ligne de commande

**Indication :** Ces commandes ne sont disponibles que sur les serveurs Linux

**Danger :** Avant d'utiliser cette méthode, assurez-vous que votre serveur WAPT est sûr et n'est pas sous le contrôle d'une entité tierce non autorisée.

- Copiez vos `.crt` et `.pem` dans `/tmp/` sur le serveur WAPT en utilisant **Winscp** ou un outil équivalent.
- Il est alors possible de re-signer tous les paquets en une seule fois sur le serveur WAPT avec les commandes suivantes.

```
wapt-signpackages -d /var/www/wapt-host -c /tmp/wapt_pub_key.crt -k /tmp/wapt_priv_key.pem -s
wapt-signpackages -d /var/www/wapt -c /tmp/wapt_pub_key.crt -k /tmp/wapt_priv_key.pem -s
wapt-scanpackages /var/www/wapt/
```

**Avertissement :** Si l'erreur **Access violation** apparaît, c'est que le paquet est trop gros.

Éditez le paquet et suivez *cette procédure*.

**Indication :** Utilisez cette méthode si la resignature à partir de la méthode de la console WAPT n'aboutit pas.

**Attention :** Supprimez vos fichiers `.crt` et `.pem` de `/tmp/` sur le serveur WAPT.

## 41.9 Utilisation des lignes de commande avec des options

Option	Définition
<code>--version</code>	Afficher le numéro de version du programme et c
<code>-h</code>   <code>--help</code>	Afficher le message d'aide et quitter
<code>-c CONFIG</code>   <code>--config=CONFIG</code>	Chemin vers un autre fichier comme <code>wapt-get.</code>
<code>-l LOGLEVEL</code>   <code>--loglevel=LOGLEVEL</code>	Niveau des fichiers journaux suivant cette liste :
<code>-D</code>   <code>--direct</code>	N'utilise pas le service http pour les mises à jour
<code>-S</code>   <code>--service</code>	Demander un utilisateur pour Waptservice
<code>-u</code>   <code>--update-packages</code>	Exécuter <code>wapt-get update</code> avant la ligne de co
<code>-f</code>   <code>--force</code>	Forcer la ligne de commande
<code>-p PARAMS</code>   <code>--params=PARAMS</code>	Configurer les paramètres comme un objet JSON
<code>-r WAPT_URL</code>   <code>-repo WAPT_URL</code>   <code>--repository=WAPT_URL</code>	Remplace l'URL du dépôt principal de wapt à pa
<code>-y</code>   <code>--hide</code>	Cacher la console pendant l'exécution de la ligne
<code>-F FILTER_ON_HOST_CAP</code>   <code>--use-host-caps=FILTER_ON_HOST_CAP</code>	Filtrer les paquets en fonction des capacités actue
<code>-i</code>   <code>--inc-release</code>	Augmentation du numéro de version lors de la co
<code>-a UPDATE_SERVER_STATUS</code>   <code>--update-server-status=UPDATE_SERVER_STATUS</code>	Envoyer le statut mis à jour de l'hôte (soft, paque
<code>keep-signature-date</code>	Conserve la date de signature du paquet et la date

suite sur la page suivante

Tableau 2 – suite de la page précédente

Option	Définition
-s SECTION_FILTER   --sections=SECTION_FILTER	Add a filter <b>section</b> to wapt-get search. Secti
-o REDIRECT_OUTPUT   --output=REDIRECT_OUTPUT	Redirige les sorties vers un fichier donné .ini
-j   -json	Switch la sortie json pour les scripts
-e ENCODING   --encoding=ENCODING	Changer l'encodage des caractères pour la sortie
-x EXCLUDES   --excludes=EXCLUDES	Liste séparée par des virgules des fichiers ou des
-k PERSONAL_CERTIFICATE_PATH   --certificate=PERSONAL_CERTIFICATE_PATH	Chemin vers le certificat PEM X509 pour signer
-w PRIVATE_KEY_PASSWD   --private-key-passwd=PRIVATE_KEY_PASSWD	Chemin d'accès au fichier contenant le mot de pa
-U USER   --user=USER	Pour définir un utilisateur interactif
-g USERGROUPS   --usergroups=USERGROUPS	Groupes de l'utilisateur final sous la forme d'un
-t MAX_TTL   --maxttl=MAX_TTL	durée axiale d'exécution en minutes du processu
-L LANGUAGE   --language=LANGUAGE	Langue prioritaire pour l'installation, exemple : f
-m MD   --message-digest=MD	Type de résumé de message pour wapt-get si
-n   --newest-only	Renvoie uniquement la version la plus récente de
--locales=LOCALES	Remplacer le filtre local des paquets pour wapt-
--maturity=MATURITY	Définir/modifier la maturité du paquet lors de la
--pin-server-cert	Lors de l'enregistrement, épinglez le certificat du
--wapt-server-url=SET_WAPTSERVER_URL	Lorsque le paramètre wapt_server n'est pas dé
--wapt-repo-url=SET_WAPTREPO_URL	Lorsque le paramètre repo_url n'est pas défini,
--wapt-server-user=WAPT_SERVER_USER	Définit l'utilisateur chargé de télécharger les paq
--wapt-server-passwd=WAPT_SERVER_PASSWD	Définit le mot de passe pour télécharger des paq
--log-to-windows-events	Consigner les étapes dans le journal des événeme
--use-gui	Force l'utilisation de GUI Helper même si elle n
--no-ide	Ne pas lancer l'ide lors de l'édition d'un paquet

Pour créer vos paquets et les personnaliser, suivez cette documentation et vous maîtriserez très vite les subtilités du fonctionnement de WAPT.

## Structure détaillée d'un paquet WAPT

Un paquet WAPT est un fichier `.zip` contenant plusieurs éléments :

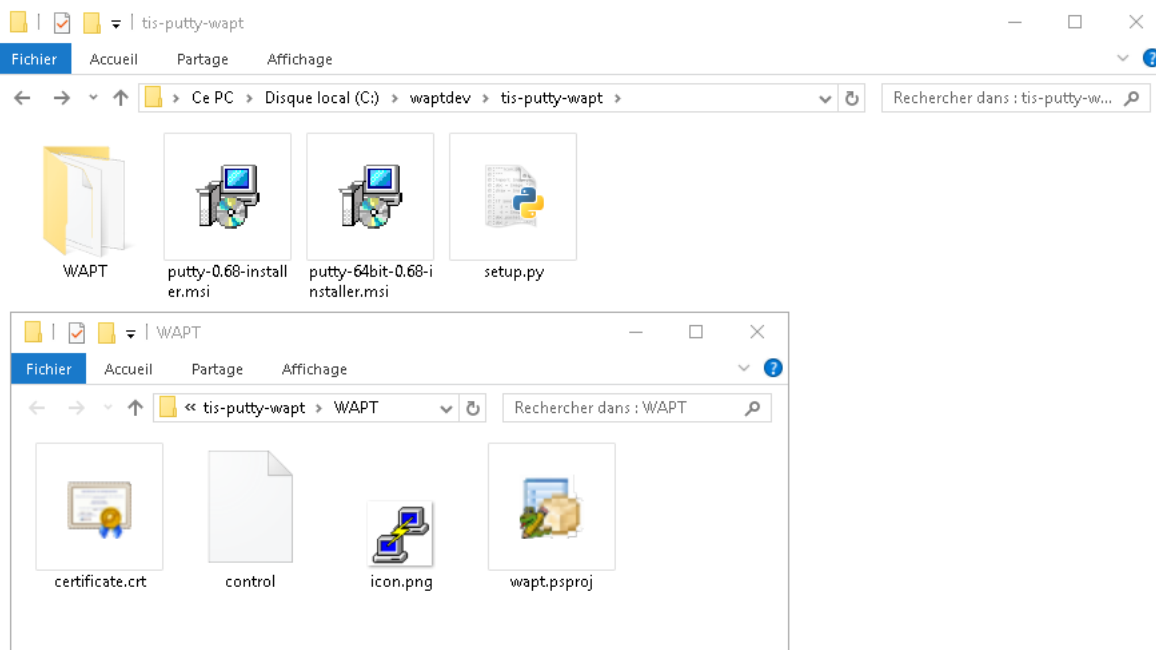


FIG. 1 – Structure d'un paquet WAPT

- un fichier `setup.py` ;
- un ou plusieurs fichier(s) binaire(s) ;
- des fichiers supplémentaires optionnels ;
- un fichier `control` dans le dossier `WAPT` ;

- un fichier `icon.png` dans le dossier `WAPT`;
- un fichier `certificate.crt` dans le dossier `WAPT`;
- un fichier `manifest.sha256` dans le dossier `WAPT`;
- un fichier `signature.sha256` dans le dossier `WAPT`;
- un fichier `wapt.pspproj` dans le dossier `WAPT`, ce fichier est utilisé pour stocker la configuration de **PyScripter** créé pour développer des paquets `WAPT`;
- depuis `WAPT 1.8`, un dossier caché `.vscode` contenant un fichier `launch.json` et un fichier `settings.json` est utilisé pour stocker la configuration de **VScode** pour développer des paquets `WAPT`;

### 42.1 Le fichier *control*

Le fichier `control` est la carte d'identité du paquet.

```
package      : tis-firefox-esr
version      : 62.0-0
architecture : all
section      : base
priority     : optional
maintainer   : Administrateur
description  : Firefox Web Browser French
description_fr : Navigateur Web Firefox Français
description_es : Firefox Web Browser
depends       :
conflicts    :
maturity     : PROD
locale       : fr
target_os    : windows
min_os_version :
max_os_version :
min_wapt_version : 1.6.2
sources      :
installed_size :
impacted_process : firefox.exe
audit_schedule :
editor       : Mozilla
keywords     : Navigateur
licence      : MPL
homepage     : https://www.mozilla.org/en-US/firefox/organizations/
signer       : Tranquil IT
signer_fingerprint: 459934db53fd804bbb1dee79412a46b7d94b638737b03a0d73fc4907b994da5d
signature    : MLOzLiz0qCHN5fChdylnvXUZ8xNJj4rEu5FAAsDTdEtQ(...)hsduxGRJpN1wLEjGRaMLBlod/p8w==
signature_date : 20170704-164552
signed_attributes : package,version,architecture,section,priority,maintainer,description,depends,
↪conflicts,maturity,locale,min_os_version,max_os_version,min_wapt_version,sources,installed_size,
↪signer,signer_fingerprint,signature_date,signed_attributes
```

TABLEAU 1 – Description des options du fichier control

Paramètres	Description	Exemple de valeur
paquet	Nom du paquet	tis-geogebra
version	La version du paquet, elle ne peut pas contenir plus de 5 délimiteurs, le dernier nombre est le numéro de version du paquet	5.0.309.0-1
architecture	Architecture du processeur	x64
section	Type de paquet (host, group, base)	base
priorité	Ordre de priorité d'installation du paquet (optionnel)	Non utilisable pour le moment
maintainer	L'auteur du paquet	Arnold Schwarzenegger <terminator@mydomain.lan>
description	La description du paquet qui apparaîtra dans la console et sur l'interface web	La Calculatrice Graphique pour les Fonctions, la Géométrie, l'Algèbre, le Calcul, les Statistiques et la 3D
description_fr	La description avec la langue localisée du paquet	Calculatrice graphique
depends	Les paquets qui doivent être <b>installés</b> avant l'installation du paquet	tis-java
conflicts	Les paquets qui doivent être <b>désinstallés</b> avant l'installation du paquet	tis-graph
maturity	Niveau de Maturité ( <i>BETA</i> , <i>DEV</i> , <i>PROD</i> )	PROD
locale	La langue de l'environnement pour le paquet	fr,en,es
target_os	Les Systèmes d'exploitation compatibles avec le paquet	windows,mac,linux
min_os_version	La version minimale de Windows pour que le paquet soit vu par l'agent WAPT	6.0
max_os_version	La version maximale de Windows pour que le paquet soit vu par l'agent WAPT	10.0
min_wapt_version	La version minimum de WAPT pour que le paquet fonctionne correctement	1.3.8
sources	Chemin vers l'emplacement SVN du paquet ( <b>source</b> command)	<a href="https://srv-svn.mydomain.lan/sources/tis-geogebra-wapt/trunk/">https://srv-svn.mydomain.lan/sources/tis-geogebra-wapt/trunk/</a>
installed_size	Espace disque libre minimum requis pour l'installation du paquet	254251008
impacted_process	Indique une liste de processus impactés lors de l'installation d'un paquet	firefox.exe
audit_schedule	La périodicité de l'exécution de la fonction d'audit dans le paquet WAPT	60
editor	L'éditeur du logiciel contenu dans le paquet	Mozilla
license	Référence de la licence du logiciel	GPLV3
keywords	Ensemble de mots-clé décrivant le paquet WAPT	Productivité, Traitement de texte
homepage	Site officiel du logiciel contenu dans le paquet WAPT	<a href="https://www.tranquil.it/">https://www.tranquil.it/</a>
signer	Le CommonName (CN) (I.E : nom commun) du signataire du paquet	Tranquil IT
signer_fingerprint	L'emprunte du propriétaire du certificat	2BA-FAF007C174A3B00F12E9CA1E74956
signature	Le hash en SHA256 du paquet	MLOz-Liz0qCHN5fChdylnvXUZ8xNj4rEu5FA...
signature_date	Date à laquelle le paquet a été signé	20180307-230413
signed_attributes	Liste des attributs des paquets qui sont signés	package, version, architecture, section, priority, maintainer, description, depends, conflicts, maturity, locale, min_wapt_version, sources, installed_size, signer, signer_fingerprint, signature_date, signed_attributes
<b>42.1. Le fichier control</b>		

**Attention :** Si le fichier control contient des caractères spéciaux, le fichier control doit être sauvegardé en format **UTF-8 (No BOM)**.

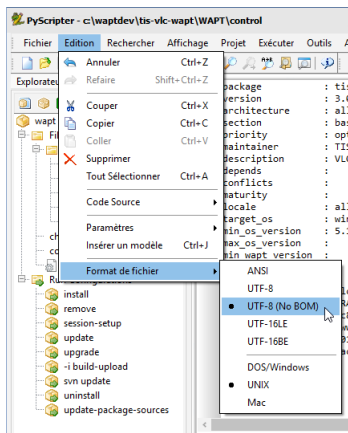


FIG. 2 – PyScripter - UTF-8 (No BOM)

## 42.1.1 Champ détails

### paquet

Le nom du paquet WAPT, sans aucun accent, ni espace, ni aucun caractère spécial ou caractère en majuscule.

### version

Il est préférable de toujours commencer avec la version du logiciel (**numéro uniquement**) séparé par des points (.) et terminant par la version du paquet WAPT séparé par un tiret (-).

### architecture

Nouveau dans la version 1.5.

Définit où les paquets peuvent être installés ; sur des postes équipés de processeurs en x64 ou x32.

---

**Note :** Un paquet en x64 sera invisible pour un agent WAPT installé sur une machine en x86.

---

Les valeurs autorisées :

- **x86** : le paquet est fait des ordinateurs en 32bits ;
- **x64** : le paquet est fait des ordinateurs en 64bits ;
- **all** : le paquet est fait des ordinateurs en 64bits ou 32bits ;



## section

- **host** : paquet hôte ;
- **group** : Paquet groupe ;
- **base** : Paquet base ;
- **unit** : Paquet OU ;

## priority

Cette option n'est actuellement pas supporté pour l'instant. Ce champ sera utilisé pour définir la priorité d'installation d'un paquet. Cette fonctionnalité sera utile pour définir des mises à jour de sécurité obligatoires.

## maintainer

Définit le créateur du paquet WAPT.

---

**Note** : Définir l'adresse mail du créateur du paquet peut être utile.

Utilisez le format Prénom NOM <email@exemple.com>.

---

## description

Décrit la fonctionnalité du paquet qui apparaîtra dans la console et sur l'interface web en local sur le <http://127.0.0.1:8088>.

---

**Indication** : Ajouter un champ `description_fr` ou `description_es` vous permet de faire un paquet dont la description est internationale. Si la langue n'existe pas, l'agent WAPT va utiliser la langue qui est par défaut.

---

## depends

Définit si des paquets doivent être installés avant, par exemple, *tis-java* est une dépendance pour que le paquet *LibreOffice*, il doit être installé avant.

Plusieurs dépendances peuvent être définies en les séparant par des virgules (,).

```
depends: tis-java,tis-firefox-esr,tis-thunderbird
```

## conflicts

Fonctionne de manière opposé à *depends*.

Les *conflicts* définissent un/des paquet(s) qui doivent être désinstallés avant d'installer un paquet, par exemple *tis-firefox* doit être désinstallé avant que le paquet *tis-firefox-esr* le soit, ou *OpenOffice* doit être supprimé avant que *LibreOffice* soit installé.

Plusieurs conflits peuvent être définis en les séparant par des virgules (,).

```
conflicts: tis-firefox
```

### maturity

Définit la maturité d'un paquet.

Par défaut, les agents WAPT verront des paquets marqués comme *PROD* et des paquets avec un maturité vide.

Pour qu'un poste puisse voir des paquets avec des niveaux de maturités différents, vous devrez configurer l'attribut *maturities* dans le `wapt-get.ini` de l'agent WAPT.

### locale

Définit la langue du paquet WAPT.

Un agent WAPT verra par défaut des paquets qui sont configurés pour son langage d'environnement ainsi que les paquets sans langues spécifiées.

Pour qu'un ordinateur voit un paquet dans une autre langue, vous devrez configurer les *locales* dans le `wapt-get.ini` de l'agent WAPT.

```
locales = fr,en,es
```

La langue remplie dans le champ doit être au format [ISO 639-1](#).

### target\_os

Définit le Système d'exploitation du paquet.

Un agent WAPT verra par défaut les paquets qui sont configurés pour son système d'exploitation et les paquets sans système d'exploitation spécifié.

Depuis la version 1.8 le champ *target\_os* peut soit être *windows*, *macos*, *linux* ou laissé vide.

### min\_os\_version

Pour un *target\_os windows*, ce champ définit la [Version du Système d'Exploitation de Windows minimale](#). Par exemple, cet attribut peut être utilisé pour éviter d'installer sur un WindowsXP des paquets qui ne fonctionnent que sur Windows7 et Plus.

Depuis la version 1.8, il peut aussi définir la version minimum de Mac OS. Nous vous recommandons de ne pas l'utiliser avec Linux car il y a plusieurs distributions différentes.

### max\_os\_version

Pour un *target\_os windows*, ce champ définit la [Version du Système d'Exploitation de Windows maximale](#). Par exemple, cet attribut peut être utilisé pour installer sur un Windows7 et Plus des paquets qui ne sont plus supportés sur des WindowsXP.

Depuis la version 1.8, il peut aussi définir la version minimum de Mac OS. Nous vous recommandons de ne pas l'utiliser avec Linux car il y a plusieurs distributions différentes.

## min\_wapt\_version

La version minimum de WAPT pour installer un paquet

---

**Note :** Les fonctionnalités de WAPT évoluant, certaines fonctions que vous êtes susceptible d'avoir utilisé dans de vieux paquets deviennent obsolete avec de nouvelles version d'agents WAPT.

---

## sources

Définit un dépôt SVN, par exemple :

— <https://svn.mydomain.lan/sources/tis-geogebra-wapt/trunk/>

Cette méthode permet de versionner un paquet de travailler collaborativement dessus.

---

**Indication :** Le versionnage de paquet est particulièrement utile lorsque plusieurs personnes créent des paquets de manière collaboratif. Cette fonction est aussi utile pour tracer l'historique d'un paquet si vous êtes sujet à des Régulations dans votre entreprise.

---

## installed\_size

Définit l'espace disque libre minimum requis pour installer le paquet.

```
installed_size: 254251008
```

Le test d'espace disque disponible est fait sur le dossier C:\Program Files.

La valeur définie dans *installed\_size* doit être en bytes.

---

**Indication :** Pour convertir les valeurs de stockage en bytes, visitez <https://bit-calculator.com/>.

---

## impacted\_process

Indique les processus qui sont impactés lors de l'installation d'un paquet.

Exemple :

```
impacted_process : firefox.exe,chrome.exe,iexplorer.exe
```

Ce champ est utilisé par la fonction **install\_msi\_if\_needed** et **install\_exe\_if\_needed** si *killbefore* n'a pas été rempli.

*impacted\_process* est aussi utilisé lors de la désinstallation du paquet. Cela permet de fermer l'application si l'application tourne avant d'être désinstallé.

### audit\_schedule

Périodicité d'exécution des contrôles de l'audit.

```
audit_schedule : 60
```

La périodicité peut être indiquée de plusieurs manières :

- un entier (en minutes);
- un entier suivi d'une lettre (*m* = minutes , *h* = heure , *d* = jour , *w* = semaine);

### editor

Nouveau dans la version 1.6.

L'éditeur logiciel du binaire inclus dans le paquet *base*.

```
editor: Mozilla
```

Les valeurs peuvent être utilisées comme filtres dans la console WAPT ainsi que le self-service.

### keywords

Nouveau dans la version 1.6.

Liste de mot-clé pour catégoriser le paquet WAPT.

```
keywords: office
```

Les valeurs peuvent être utilisées comme filtres dans la console WAPT ainsi que le self-service.

### license

Nouveau dans la version 1.6.

Fait référence à la licence des binaires du logiciel contenus dans le paquet.

```
license : GPLv3
```

Les valeurs peuvent être utilisées comme filtres dans la console WAPT ainsi que le self-service.

### homepage

Nouveau dans la version 1.6.

Site officiel des binaires du logiciel dans le paquet WAPT.

```
homepage : https://wapt.fr
```

Les valeurs peuvent être utilisées comme filtres dans la console WAPT ainsi que le self-service.

**signer**

Automatiquement complété durant la signature du paquet.

le CN du certificat. Habituellement, c'est le nom complet du signataire.

**signer\_fingerprint**

Automatiquement complété durant la signature du paquet.

Emprunte de la clé privée du signataire du paquet.

**signature**

Automatiquement complété durant la signature du paquet.

Signature des attributs dans le paquet.

**signature\_date**

Automatiquement complété durant la signature du paquet.

Date à laquelle les attributs du paquet ont été signés.

**signed\_attributes**

Automatiquement complété durant la signature du paquet.

Liste des attributs signés dans le paquet.

## 42.2 Le fichier *setup.py*

### 42.2.1 import setuphelpers

Cette ligne se trouve au début de chaque paquet WAPT qui embarque un `setup.py` :

```
from setuphelpers import *
```

Le paquet import toutes les fonctions du `setuphelpers`.

Le *Setuphelpers* est une librairie WAPT qui offre un ensemble de méthodes afin de développer plus facilement des paquets entièrement fonctionnels.

### 42.2.2 uninstallkey list

Nous trouvons alors :

```
uninstallkey = ['tisbnaps2', 'Mozilla Firefox 45.6.0 ESR (x86 fr)']
```

Ici, nous associons une liste de *uninstall keys* (clé de désinstallation) au paquet. Lorsqu'un paquet est supprimé, l'agent WAPT va regarder l'*uninstallkey* contenu dans la base de registre associée au paquet. Cette *uninstallkey* va indiquer à WAPT les actions à lancer avant de supprimer le logiciel.

Même s'il n'y a pas d'*uninstallkey* pour un logiciel, il est nécessaire de déclarer une liste *uninstallkey* vide :

```
uninstallkey = []
```

### 42.2.3 Fonction install()

Viens alors la déclaration des fonctions dans `setup.py`.

C'est la recette du paquet WAPT, l'ensemble d'instructions qui sera exécuté.

```
def install():  
    run('"install.exe" /S')
```

## 42.3 Le fichier *wapt.psproj*

Le fichier `wapt.psproj` est le projet du paquet qui se trouve dans le dossier WAPT.

C'est le fichier de projet de **PyScripter** pour le paquet WAPT.

Pour éditer le paquet avec **PyScripter**, ouvrez juste le fichier.

## 42.4 Le fichier *icon.png*

Le fichier icône `icon.png` est situé dans le dossier WAPT.

Il associe une icône au paquet.

Cette icône apparaîtra sur l'interface web locale ou le WAPT self-service (<http://127.0.0.1:8088>).

---

**Indication :** L'icône doit être un fichier 48px par 48px.

---

## 42.5 Le fichier *manifest.sha256*

Le fichier manifeste `manifest.sha256` est situé dans le dossier WAPT.

Il contient l’empreinte en sha256 de tous les fichiers du paquet WAPT.

## 42.6 Le fichier *signature*

Le fichier `signature` est situé dans le dossier WAPT.

Il contient la signature du fichier `manifest.sha256`.

Lorsqu’un paquet s’installe, **wapt-get** vérifie :

- que la signature du `manifest.sha256` corresponde à l’actuel fichier `manifest.sha256` (l’agent va vérifier les certificats publics dans `C:\Program Files (x86)\wapt\ssl` sur Windows et `/opt/wapt/ssl` sur Linux et MacOS);
- que l’empreinte sha256 de chaque fichier est identique à l’empreinte dans le fichier `manifest.sha256`;

## 42.7 Autres fichiers

D’autres fichiers peuvent être embarqués dans le paquet WAPT. Par exemple :

- un installeur en plus de votre fichier `setup.py` pouvant être appelé dans votre **setup.py**;
- un fichier de réponse à passer à l’installeur du logiciel;
- un fichier de licence;





## De Python 2 à Python3

**Attention :** Avec WAPT 2.0, le fonctionnement interne de WAPT est passé à python3. Les paquets WAPT doivent aussi suivre la nouvelle syntaxe python3.

TABLEAU 1 – Les principales différences de syntaxe

Syntaxe	Python 2	Python 3
print	print 'Hello'	print('Hello')
unicode string	ur	r
opérateurs	<> <=> !=	!=
Accès à la base de registre Windows	_winreg	winreg

**Indication :** Pour plus de détails sur les changements, consultez :

- [https://python-future.org/compatible\\_idioms.html](https://python-future.org/compatible_idioms.html)
- <https://blog.couchbase.com/tips-and-tricks-for-upgrading-from-python-2-to-python-3/>



---

## Créer son environnement de développement de paquets WAPT

---

### 44.1 Pré-requis

**Attention :**

- Il est **impératif** d'être en possession d'un compte *Administrateur Local* de la machine pour cette opération.
- Nous vous conseillons de créer / éditer vos paquets dans un environnement maîtrisé, sain et *jetable*.
- L'utilisation d'une machine virtuelle autonome (type Virtualbox ou équivalent) est vivement recommandée.
- Importer le paquet *tis-pyscripter* dans votre dépôt local et l'installer sur votre machine de développement.

### 44.2 Préconisations concernant l'environnement de test

La méthode préconisée pour tester correctement vos paquets est d'utiliser un échantillon de machines représentatif de votre parc. Donc plus votre parc est hétérogène, plus votre échantillon devra être large.

Cette démarche vise à confronter le paquet WAPT à une multitude de plateformes et d'environnements afin qu'il devienne le plus abouti possible en régime de test, avant d'être basculé en production.

## 44.3 Démarche de test

### 44.3.1 Systèmes d'exploitation et architectures

- Windows XP;
- Windows 7;
- Windows 10;
- Windows Server 2008 R2;
- Windows Server 2012;
- x86;
- x64;
- Machine physique et virtuelle;
- ordinateurs portables;

---

**Indication :** On testera si possible les versions RC / Beta des OS si elles sont disponibles (exemple : Windows 10 Creators Update).

---

### 44.3.2 L'état des mises à jour Windows

- **Un poste Microsoft Windows sans aucune mise à jour Windows Update :** l'objectif est de détecter les mises à jour indispensables au bon fonctionnement du logiciel et adapter le paquet en conséquence;
- **Un poste Microsoft Windows à jour avec les toutes dernières MàJ Windows Update :** l'objectif est de détecter les mise à jour en conflit avec le logiciel et d'adapter le paquet en conséquence;

### 44.3.3 Etat des installations des logiciels

- **Un poste avec peu de logiciels déjà installés :** l'objectif est de détecter une dépendance possible à Java ou autre framework applicatif;
- **Les postes avec beaucoup de logiciels déjà installés :** l'objectif est de détecter un conflit avec une application existante;
- **Installer les anciennes versions du logiciel :** il est possible que l'installeur ne supporte pas l'écrasement d'une installation précédente, dans ce cas il faudra prévoir la désinstallation des anciennes versions avant d'installer la nouvelle version;

---

## Les principes de création d'un paquet WAPT à partir d'un modèle depuis la console

---

---

**Indication :** Pour créer des paquets à partir de la console, il faut d'abord avoir installé l'environnement de développement WAPT *tis-pyscripter*.

---

### 45.1 Créer un paquet WAPT depuis la console

Dans cet exemple, l'installateur de 7zip est utilisé au format MSI.

- [Télécharger 7-zip MSI x64](#) .
- Créer le modèle de paquet depuis l'installateur.  
Dans la console WAPT, cliquer sur *Outils* → *Créer un modèle de paquet depuis un installateur* :  
Sélectionner l'installateur MSI téléchargé et renseigner les différentes informations demandées. Veillez bien à ce que le nom du paquet ne contienne pas de numéro de version.
- Deux solutions sont proposées :
  - Cliquer sur *Créer et éditer* (recommandée) pour lancer la personnalisation du paquet et l'adapter aux besoin spécifique de votre Organisation.
  - Cliquer sur *Créer et Téléverser* pour lancer la création et le chargement direct du paquet sur le serveur WAPT (non recommandé).

**Attention :** Le bouton *Build and upload* envoi directement le paquet dans le dépôt privé sans tester l'installation. Cette méthode fonctionne assez bien avec les MSI car leur installation est standard. Cependant la deuxième méthode qui consiste à tester localement le paquet d'abord puis à l'uploader est la méthode recommandée.

---

**Note :** Une ancienne méthode en ligne de commande est disponible [ici](#).

---

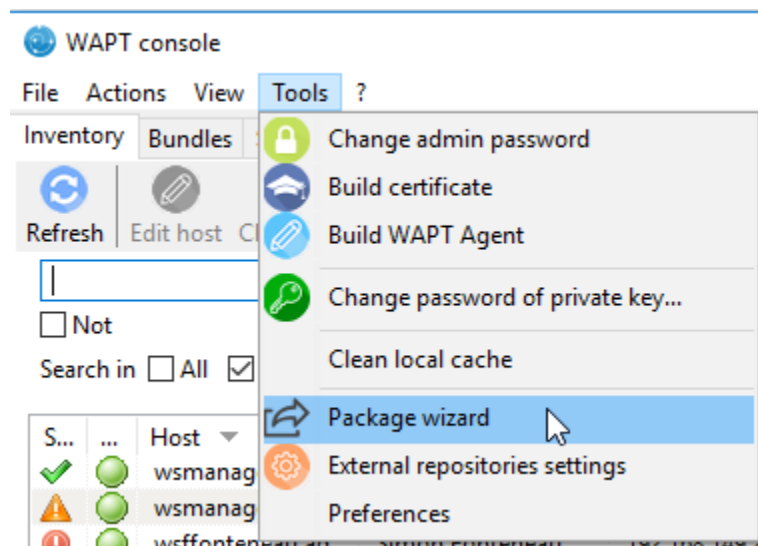


FIG. 1 – PyScripter - Menu pour la création de modèle de paquets depuis la console

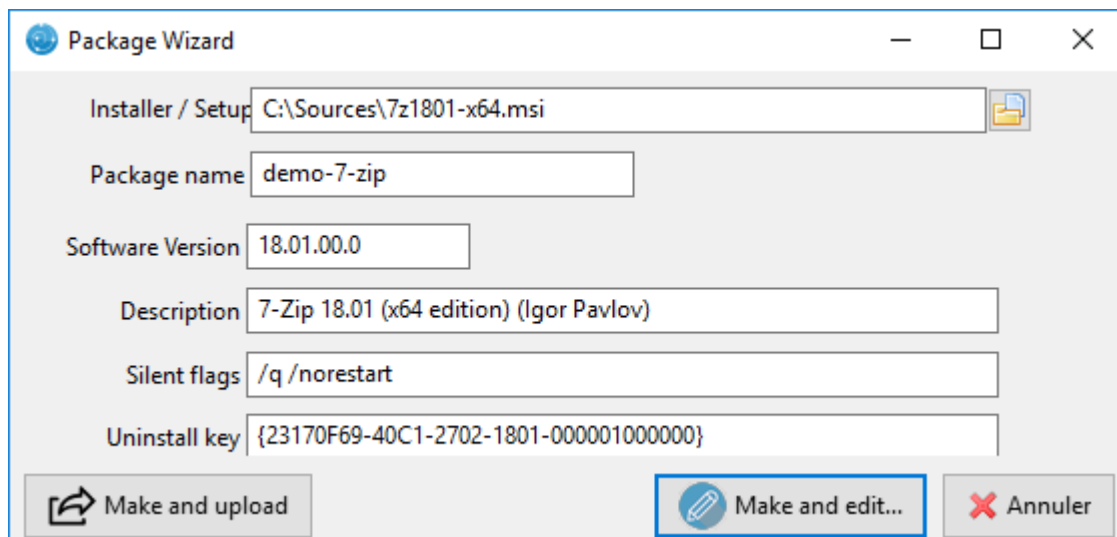


FIG. 2 – PyScripter - Renseignements nécessaires pour la création du paquet

## 45.2 Personnaliser le paquet avant de le téléverser dans votre dépôt

La méthode conseillée avant l'upload d'un paquet est de personnaliser son comportement en l'éditant avec **PyScripter**.

Lors de la création du modèle de paquet, cliquer sur *OK*.

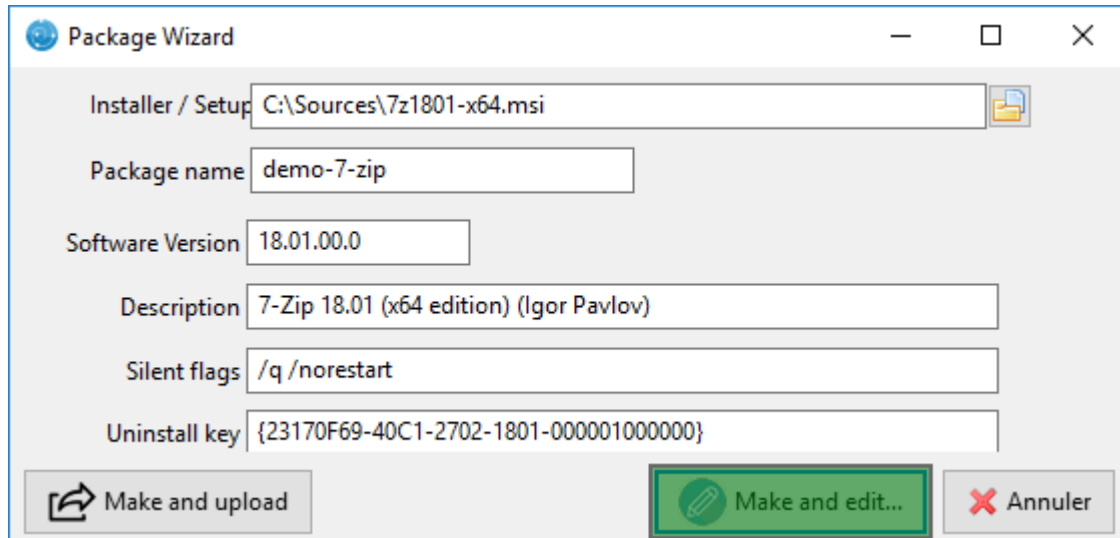


FIG. 3 – PyScripter - Renseignements nécessaires pour la création du paquet

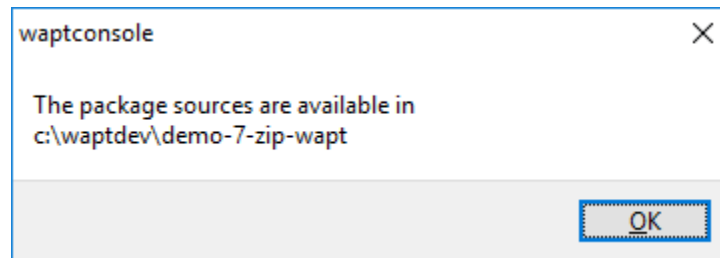


FIG. 4 – PyScripter - Validation de la création du modèle et ouverture de PyScripter

L'IDE **PyScripter** se lance et permet d'éditer les fichiers du paquet.

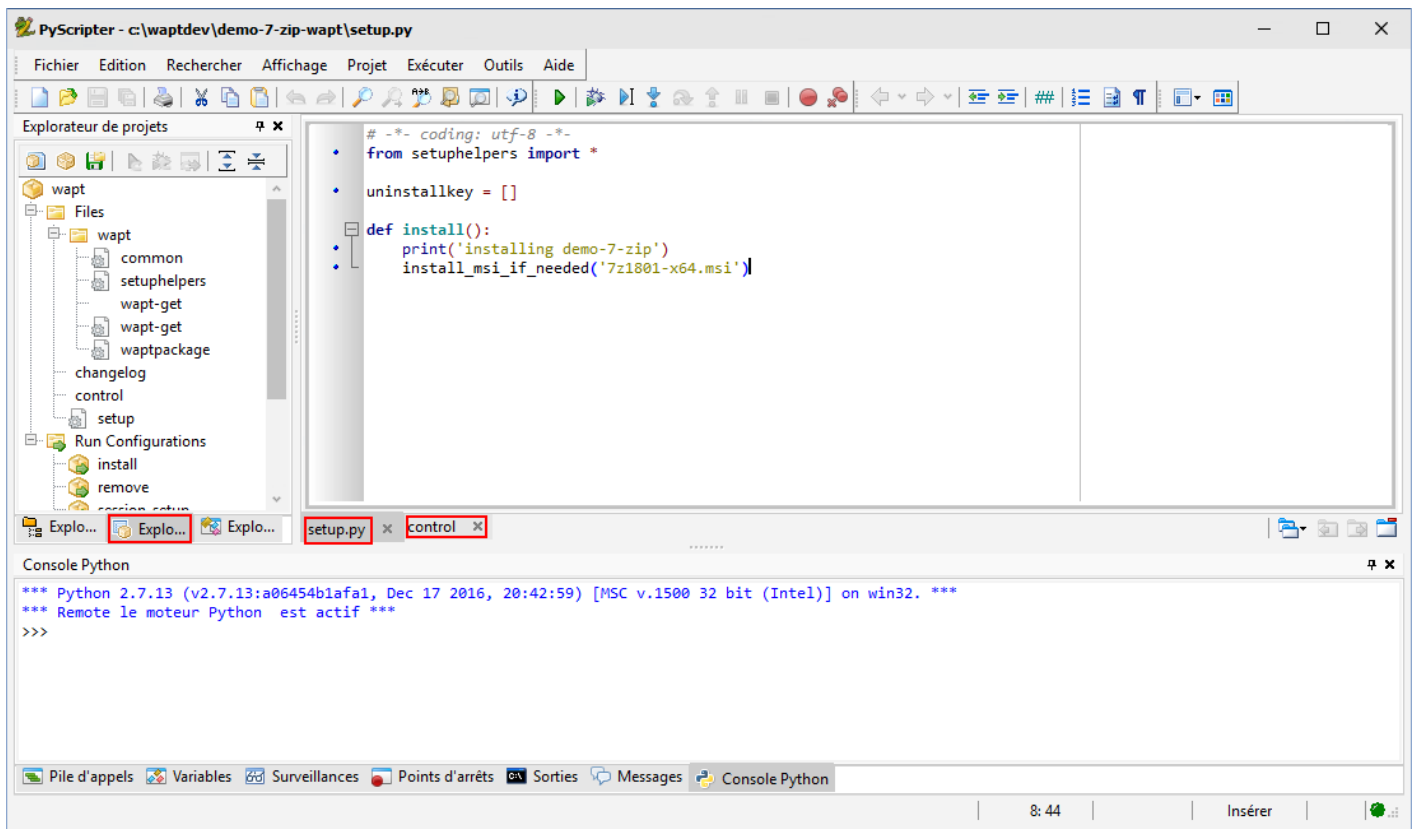


FIG. 5 – PyScripter - Personnalisation du paquet avec PyScripter



### 46.1 L'explorateur de projets PyScripter

L'explorateur de projets PyScripter liste les différents fichiers dont vous pouvez avoir besoin, notamment le fichier `control` et le fichier `setup.py`.

### 46.2 Run Configurations

Les options de **Run** dans l'explorateur de projets de **PyScripter** vont vous permettre de lancer des actions de votre paquet en cours d'édition.

### 46.3 Zone d'édition

La Zone d'édition de **PyScripter** permet d'éditer le fichier `setup.py` ainsi que le fichier `control`.

### 46.4 Console Python

C'est la console python visible dans **PyScripter**, elle va vous permettre d'afficher la sortie python lorsque vous exécuterez des commandes **run**.

Vous pouvez également l'utiliser pour tester / déboguer des portions de votre script `setup.py`.

Pour en savoir plus sur la composition d'un paquet wapt, visitez la documentation sur la structure d'un paquet WAPT.

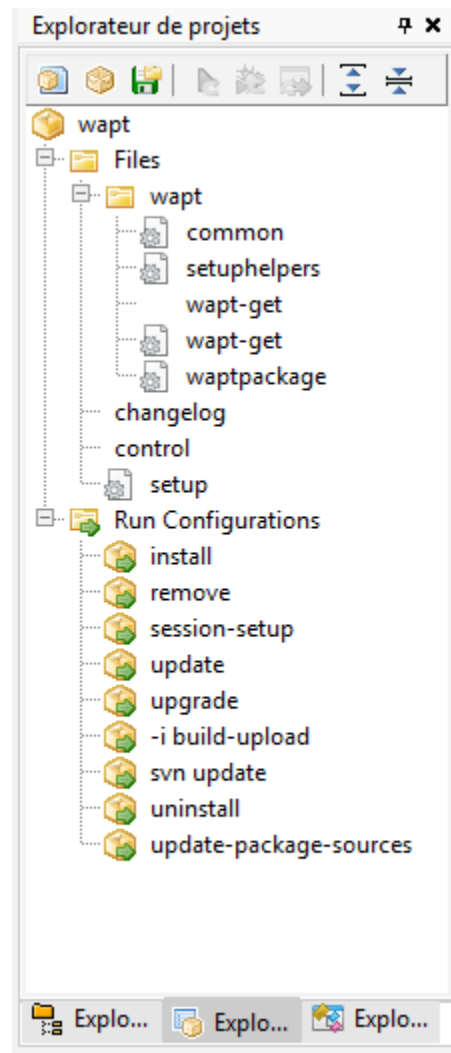


FIG. 1 – PyScripter - Explorateur de projets PyScripter

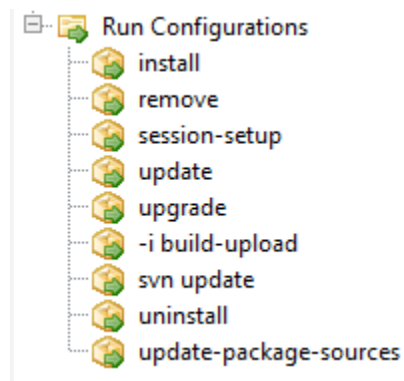


FIG. 2 – PyScripter - Commandes Run dans l'explorateur de projets PyScripter

```

• from setuphelpers import *
•
• uninstallkey = []
• vlcrc = makepath(programfiles,'VideoLAN','VLC','vlcrc')
•
• def install():
•     print("installing VLC")
•     versionpaquet = control['version'].split('-',1)[0]
•     if iswin64():
•         install_exe_if_needed('vlc-%s-win64.exe' % versionpaquet,silentflags='/S --no-qt-privacy-ask --no-qt-updates-notif',
•
•     else:
•         install_exe_if_needed('vlc-%s-win32.exe' % versionpaquet,silentflags='/S --no-qt-privacy-ask --no-qt-updates-notif',
•
•     filecopyto("vlcrc",vlcrc)
•     remove_desktop_shortcut('VLC media player')

```

FIG. 3 – PyScripter - zone d'édition de PyScripter

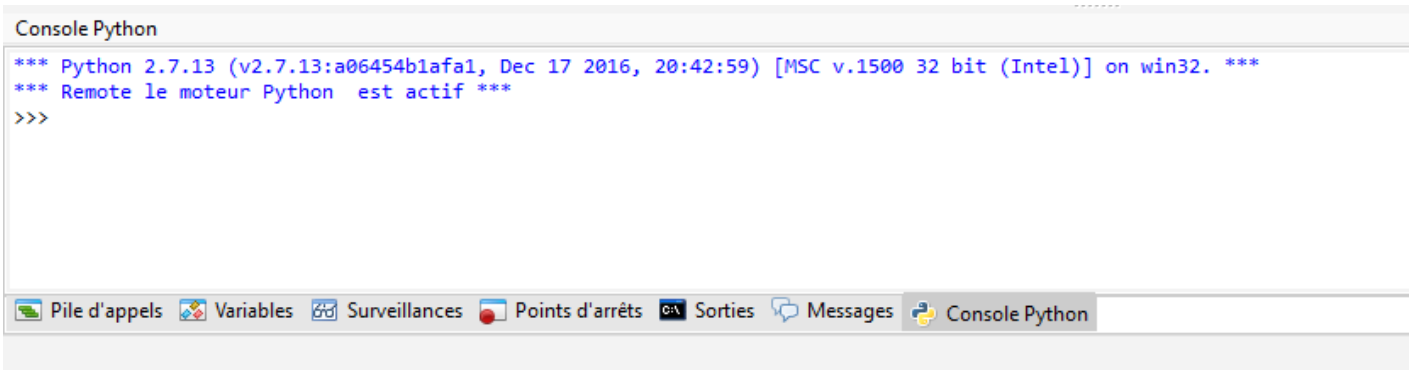


FIG. 4 – PyScripter - console python de PyScripter

## 46.5 Tester localement l'installation du paquet WAPT

Vous pouvez ensuite tester le lancement d'une installation sur votre station de développement.

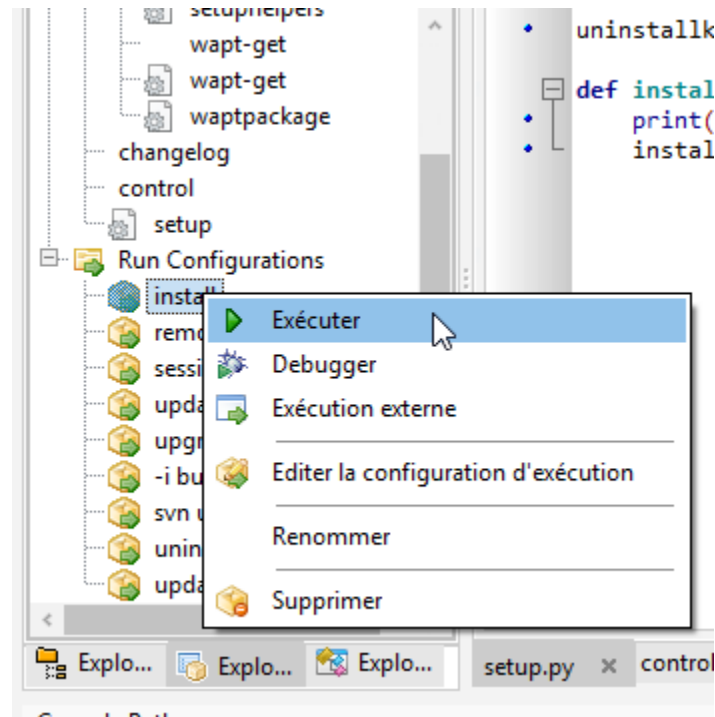


FIG. 5 – Exécuter une commande d'installation depuis la console PyScripter

La Console PyScripter vous permet de vérifier si l'installation s'est bien déroulée.

## 46.6 Tester localement la désinstallation du paquet WAPT

Vous pouvez ensuite tester le lancement d'une désinstallation sur votre station de développement.

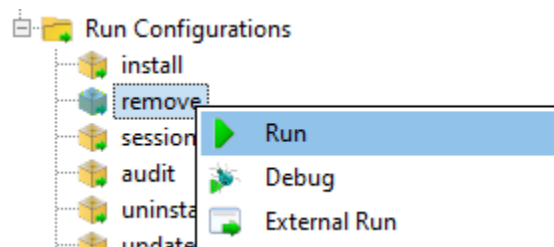


FIG. 6 – Exécuter une commande de désinstallation depuis la console PyScripter

La Console PyScripter vous permet de vérifier si l'installation s'est bien déroulée.

---

## Packager des .msi (exemple)

---

Pour cet exemple, nous prendrons **tightvnc**.

Vous pouvez le télécharger ici : <https://www.tightvnc.com/download.php>

Maintenant, vous pouvez générer votre modèle de paquet, veuillez vous référer à la *documentation pour la création de paquets à partir de la console WAPT*.

Editer le fichier control (architecture, impacted\_process, target\_os, description, maintenir ...), veuillez vous référer à la structure du fichier documentation du fichier control.

Votre **pyscripter** s'ouvre, allez dans votre setup.py

```
# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = []

def install():
    print('installing tis-tightvnc')
    install_msi_if_needed('tightvnc-2.8.5-setup-64bit.msi')
```

- La fonction testera également si une version du logiciel est déjà installée sur la machine avec la *clé de désinstallation*.
- Si présence il y a, l'installation sera enclenchée uniquement si la version actuellement installée est plus ancienne.
- Après installation, la fonction testera finalement la présence de la clé de désinstallation et sa version pour vérifier que tout s'est bien passé.

TABLEAU 1 – Liste des arguments disponibles avec *install\_exe\_if\_need*

Paramètres	Valeur par défaut	Description
msi		nom du fichier <i>.msi</i> à exécuter.
min_version	None	version minimale au dessus de laquelle il mettra à jour.
killbefore	[]	liste des programmes à tuer avant de lancer l'installation.
accept_returncodes	[0,3010]	codes de retour autres que 0 ou 3010 acceptés en retour par la fonction.
timeout	300	durée d'attente maximale d'installation (en secondes).
properties	{}	propriétés supplémentaires à passer en argument au MSI pour l'installation.
get_version	None	valeur passée en paramètre pour le contrôle de version au lieu de celle retournée par la fonction <i>installed_softwares</i>
remove_old_version	False	supprime automatiquement une ancienne version d'un logiciel dont la <i>uninstallkey</i> est identique
force	False	force l'installation du logiciel même si une <i>uninstall key</i> avec une version identique est trouvée

**Note :** La fonction **install\_msi\_if\_needed** récupère la clé de désinstallation depuis le MSI, il n'est pas nécessaire de l'écrire dans le fichier `setup.py`.

**Indication :** Vous n'avez pas non plus à remplir le champ `killbefore` si la valeur indiquée dans le champ `impacted_process` du fichier `control` est correcte.

**Note :** Le `setup.py` aurait pu ressembler à cela, mais la méthode est moins élégante car elle fait moins de vérifications :

```
# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = ["{8B9896FC-B4F2-44CD-8B6E-78A0B1851B59}"]

def install():
    print('installing tis-tightvnc')
    run('msiexec /norestart /q /i "tightvnc-2.8.5-setup-64bit.msi"')
```

Lancez l'installation et voyez ce qui se passe lorsque le logiciel est déjà installé.

```
wapt-get -ldebug install C:\waptdev\tis-tightvnc-wapt
Installing WAPT file C:\waptdev\tis-tightvnc-wapt
MSI tightvnc-2.8.5-gpl-setup-64bit.msi already installed. Skipping msiexec

Results:

=== install packages ===
C:\waptdev\tis-tightvnc-wapt | tis-tightvnc (2.8.5.0-1)
```

## 47.1 Ajouter des propriétés supplémentaires en argument

Pour ajouter des propriétés supplémentaires on va les stocker dans un élément *dict*.

```
# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = []

properties = {
    'SERVER_REGISTER_AS_SERVICE':0,
    'SERVER_ADD_FIREWALL_EXCEPTION':0,
}

def install():
    print(u'Installation en cours de TightVNC')
    install_msi_if_needed('tightvnc-2.8.5-setup-64bit.msi', properties = properties )
```

**Note :** Le `setup.py` aurait pu ressembler à cela, mais la méthode est moins élégante car elle fait moins de vérifications :

```
# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = [{"8B9896FC-B4F2-44CD-8B6E-78A0B1851B59"}]

def install():
    print('installing tis-tightvnc')
    run('msiexec /norestart /q /i "tightvnc-2.8.5-setup-64bit.msi" SERVER_REGISTER_AS_
↪SERVICE=0 SERVER_ADD_FIREWALL_EXCEPTION=0')
```

## 47.2 Vidéo de démonstration

<https://youtu.be/Z6wr6emPGCU>





## Packager des .exe (exemple)

- Télécharger l'installateur .exe à partir d'une source fiable.  
Téléchargez l'installateur au format exe Firefox ESR x64 sur <https://download.mozilla.org/?product=firefox-esr-latest-ssl&os=win64>
- Rechercher la documentation associée pour les flags silencieux :
  - Sur le site de la [Fondation Mozilla](#) .
  - Autres méthodes pour récupérer le flag silencieux.
    - Dépôt de paquets WPKG .
    - Dépôt de paquets Chocolatey <<https://chocolatey.org/packages/FirefoxESR>>`\_ .
    - Recherche Internet avec le terme *Firefox silent install*.
- Puis générer votre modèle de paquet, veuillez vous référer à la *documentation pour la création de paquets à partir de la console WAPT*. **PyScripter** se charge et ouvre le projet de paquet .exe.
- Editer le fichier control (architecture, impacted\_process, target\_os, description, maintenir ...), veuillez vous référer à la structure du fichier documentation du fichier control.
- Vérifier le contenu du fichier control. Mozilla Firefox-ESR ne répond pas aux standards et retourne un numéro de version erroné (il s'agit du numéro de version du logiciel qui créé l'installateur).
- Fichier control d'origine.

```

package      : tis-firefox-esr
version      : 4.42.0.0-0
architecture : all
section      : base
priority     : optional
maintainer   : user
description  : automatic package for firefox setup 52.6.0esr
impacted_process :
```

- Fichier control modifié :

```

package      : tis-firefox-esr
version      : 52.6.0-1
```

(suite sur la page suivante)

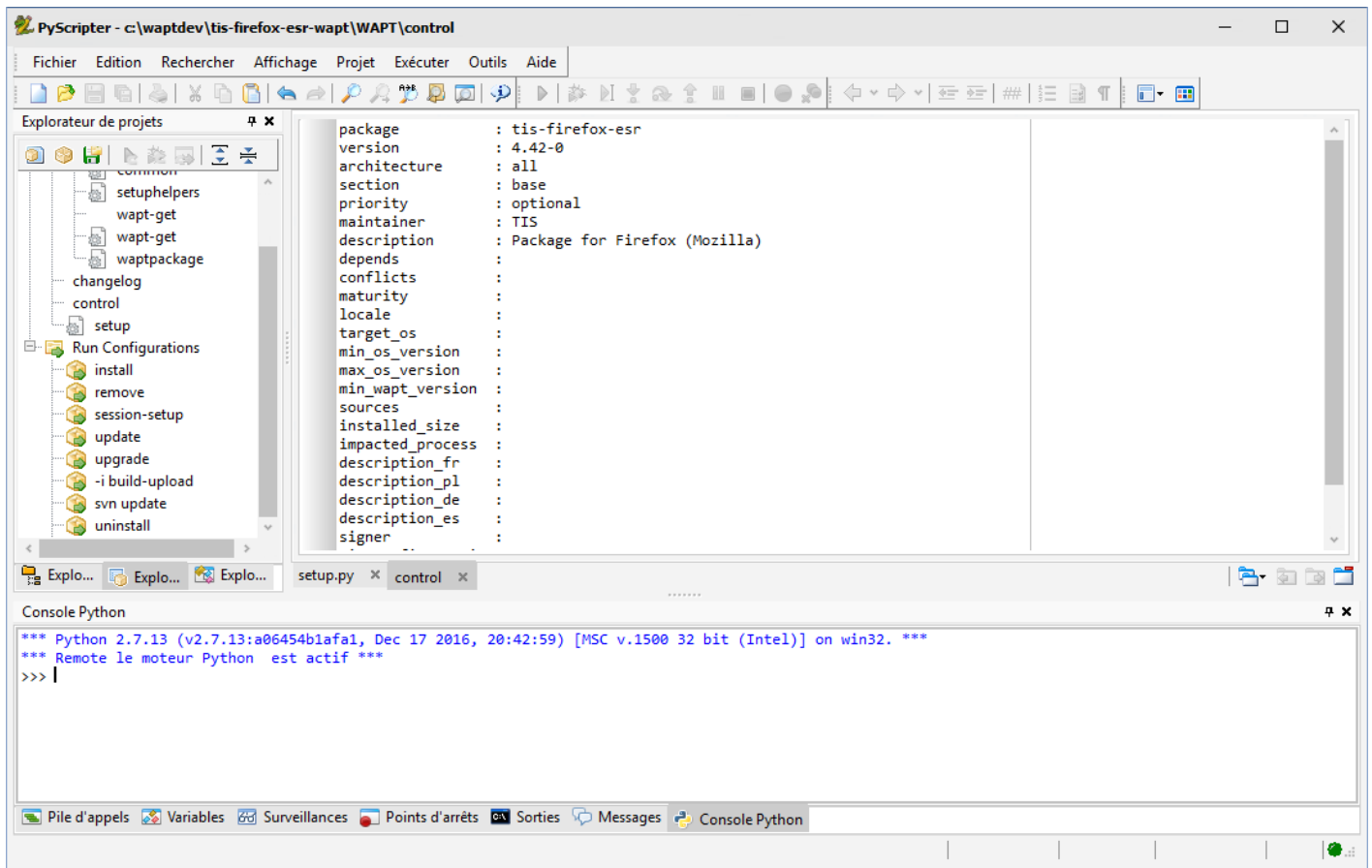


FIG. 1 – PyScripter affichant le fichier *control*

(suite de la page précédente)

```

architecture      : all
section           : base
priority          : optional
maintainer        : Tranquil-IT Systems
description       : Mozilla Firefox 52.6.0 ESR
impacted_process  : firefox.exe

```

**Note :** Il est à noter qu'une sous-version *-l* a été ajoutée au numéro de version du logiciel ; il s'agit de la version de packaging du paquet WAPT.

Il permet au développeur de paquets de publier plusieurs versions de paquets WAPT d'un même logiciel, ce qui est très utile pour un développement très rapide et itératif.

Utiliser `install_exe_if_needed`

La fonction est sensiblement la même que celle utilisée pour les installeurs `.msi`, avec quelques différences :

- La fonction nécessite l'ajout des flags silencieux en paramètre.
- La fonction nécessite l'ajout de la clé de désinstallation en paramètre.

```

# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = []

def install():
    print('installing tis-firefox-esr')
    install_exe_if_needed("Firefox Setup 45.5.0esr.exe", silentflags="-ms", key='', min_version="4.42.
→0.0")

```

TABLEAU 1 – Liste des arguments disponibles avec `install_exe_if_need`

Paramètres	Valeur par défaut	Description
<code>exe</code>		Nom du fichier <code>.exe</code> à exécuter.
<code>silentflags</code>		Paramètres silencieux à passer en argument à l'installateur.
<code>key</code>	None	Clé de désinstallation du programme.
<code>min_version</code>	None	La version minimale au dessus de laquelle il mettra à jour.
<code>killbefore</code>	[ ]	Liste des programmes à tuer avant de lancer l'installation.
<code>accept_returncodes</code>	[0,3010]	Codes de retour autres que 0 ou 3010 acceptés en retour par la fonction.
<code>timeout</code>	300	Durée d'attente maximale d'installation (en secondes).
<code>get_version</code>	None	Valeur passée en paramètre pour le contrôle de version au lieu de celle retournée par la fonction <code>installed_softwares</code> .
<code>remove_old_version</code>	False	Supprimer automatiquement une ancienne version d'un logiciel dont la clé de désinstallation est identique
<code>force</code>	False	Force l'installation même si une <code>uninstallkey</code> avec une version identique est trouvée

Le paquet aura alors ce comportement :

- Le logiciel Firefox s'installera uniquement si le logiciel n'est pas installé et si la version est strictement inférieure à 45.5.0, sauf si l'option `--force` est indiquée lors de l'installation du paquet.

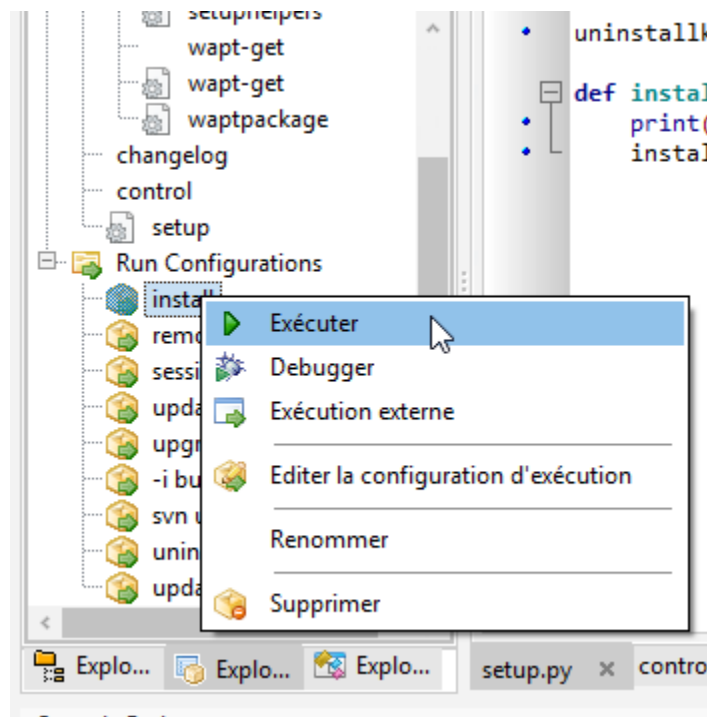
- A l'installation, les processus **firefox.exe** en cours d'exécution seront tués (avec la valeur indiquée dans `impacted_process` du fichier `control`).
- La fonction ajoutera elle-même la clé de désinstallation, donc laisser l'argument `uninstallkey` vide.
- A la fin de l'installation, la fonction ira vérifier si l'`uninstallkey` est bien présente sur le poste et si la version est bien égale ou supérieure à 45.5.0, si ce n'est pas le cas, elle basculera le paquet en **ERROR**.

## 48.1 Trouver la clé de désinstallation

Contrairement aux fichiers `.msi`, la clé pour désinstaller un `.exe` n'est pas dans les propriétés du fichier.

Vous devez donc d'abord installer le logiciel pour connaître la clé de désinstallation.

Vous devez donc démarrer une fois l'installation à partir de **pyscripter** avec le `run configuration` et ensuite `install`.



Une fois le logiciel installé, allez à la console WAPT, puis trouvez votre machine de développement.

Dans l'onglet inventaire des logiciels, trouvez votre logiciel et copiez la valeur indiquée dans la colonne clé de désinstallation.

Vous devez également vérifier la valeur de la version avec la valeur indiquée dans `min_version` dans votre `setup.py`.

Modifier votre fichier `setup.py` avec les nouveaux paramètres :

```
# -*- coding: utf-8 -*-
from setphelpers import *

uninstallkey = []

def install():
    print('installing tis-firefox-esr')
```

(suite sur la page suivante)

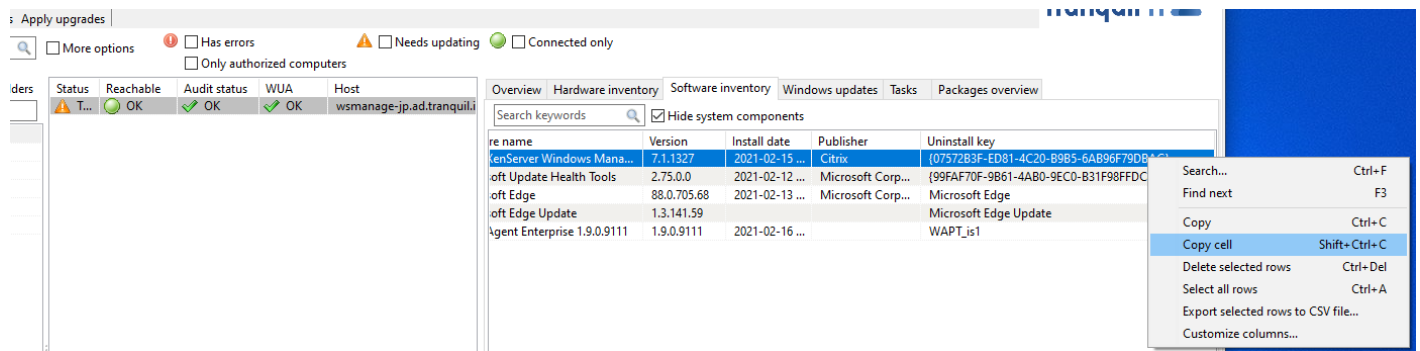
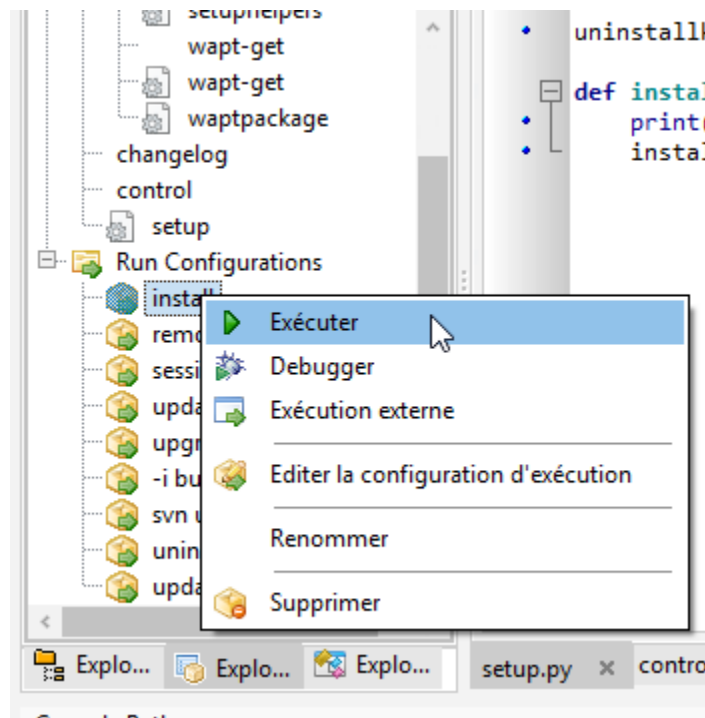


FIG. 2 – Récupérer une clé de désinstallation depuis la console

(suite de la page précédente)

```
install_exe_if_needed("Firefox Setup 45.5.0esr.exe", silentflags="-ms", key='Mozilla Firefox 45.5.0 ESR (x64 fr)', min_version="45.5.0")
```

Pour tester que votre clé fonctionne correctement, vous devez relancer une installation dans **psyscripter**.



WAPT ne tentera pas d'installer le logiciel car il est déjà présent, le message suivant devrait donc s'afficher :

```
>>>
*** Remote Interpreter Reinitialized ***
Command Line : install "c:\waptdev\tis-firefox-esr_x64_PROD_fr-wapt\WAPT\.."
Using config file: C:\Program Files (x86)\wapt\wapt-get.ini
Installing WAPT files c:\waptdev\tis-firefox-esr_x64_PROD_fr-wapt
```

(suite sur la page suivante)

```
Exe setup Firefox_Setup_78.7.1esr.exe already installed. Skipping
```

Results:

```
=== install packages ===
c:\waptdev\tis-firefox-esr_x64_PROD_fr-wapt | tis-firefox-esr (78.7.1-102)
```

Vous pouvez maintenant tester la désinstallation :

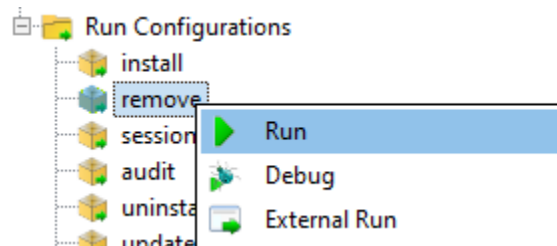


FIG. 3 – Tester la désinstallation

Vous pouvez maintenant construire et envoyer votre paquet, veuillez vous référer à la *documentation pour construire et envoyer des paquets depuis la console WAPT*.

**Note :** Si vous laissez la clé de désinstallation vide, la désinstallation de votre paquet ne fonctionnera pas.

## 48.2 Cas particulier d'un dé-installeur non-silencieux

Dans certains cas particuliers, un paquet utilisant `install_exe_if_needed` remplit la *clé de désinstallation*, mais la *clé de désinstallation* pointe vers un dé-installeur non silencieux.

Il nous faut contourner le problème en utilisant une fonction qui va supprimer la clé de désinstallation à la fin de l'installation.

```
:emphasize-lines: 13
# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = []

def install():
    install_exe_if_needed("setup.exe",
                        silentflags="/s",
                        key='{D9E87643-0005-447E-9111-78697A9C1595}',
                        min_version="14.0")
    uninstallkey.remove('{D9E87643-0005-447E-9111-78697A9C1595}')

def uninstall():
    run(r'"C:\Program Files\Kutl\uninstall.exe" /supersilent')
```

**Indication :** La fonction de désinstallation peut également être utilisée pour exécuter du code en plus de la désinstallation de logiciels, ex : supprimer un dossier, supprimer un raccourci ...

---

## 48.3 Vidéo de démonstration

[https://youtu.be/z\\_EN2CBCTcY](https://youtu.be/z_EN2CBCTcY)





## CHAPITRE 49

---

### Packager des paquets linux simples

---



---

## Construire le paquet et l'envoyer au serveur WAPT

---

— Une fois que le paquet est prêt, le construire et l'envoyer au serveur WAPT, dans la console WAPT.

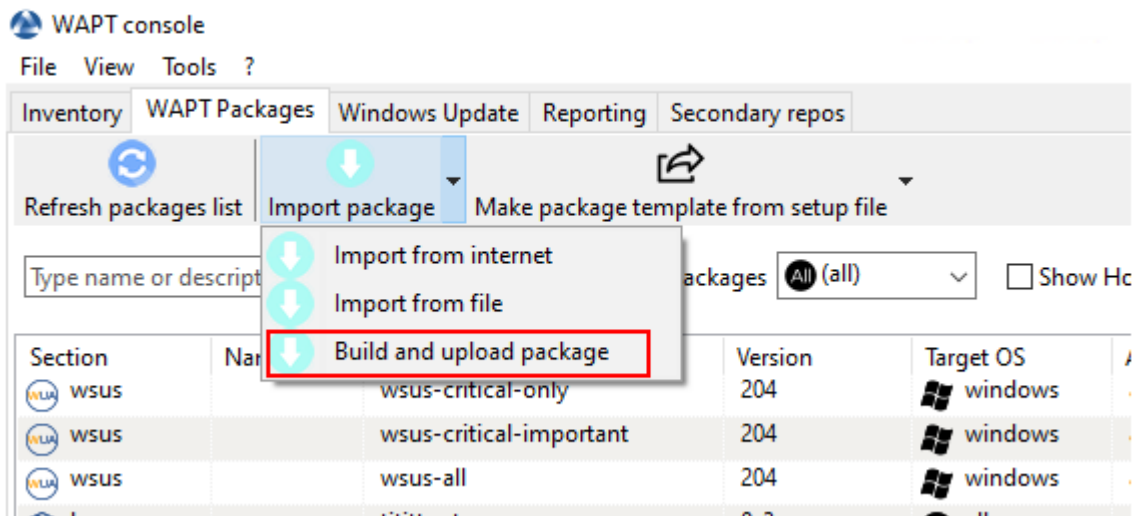


FIG. 1 – Option « Construire et charger le paquet » dans la console WAPT

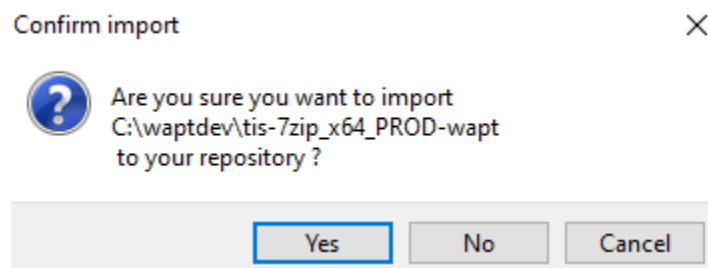
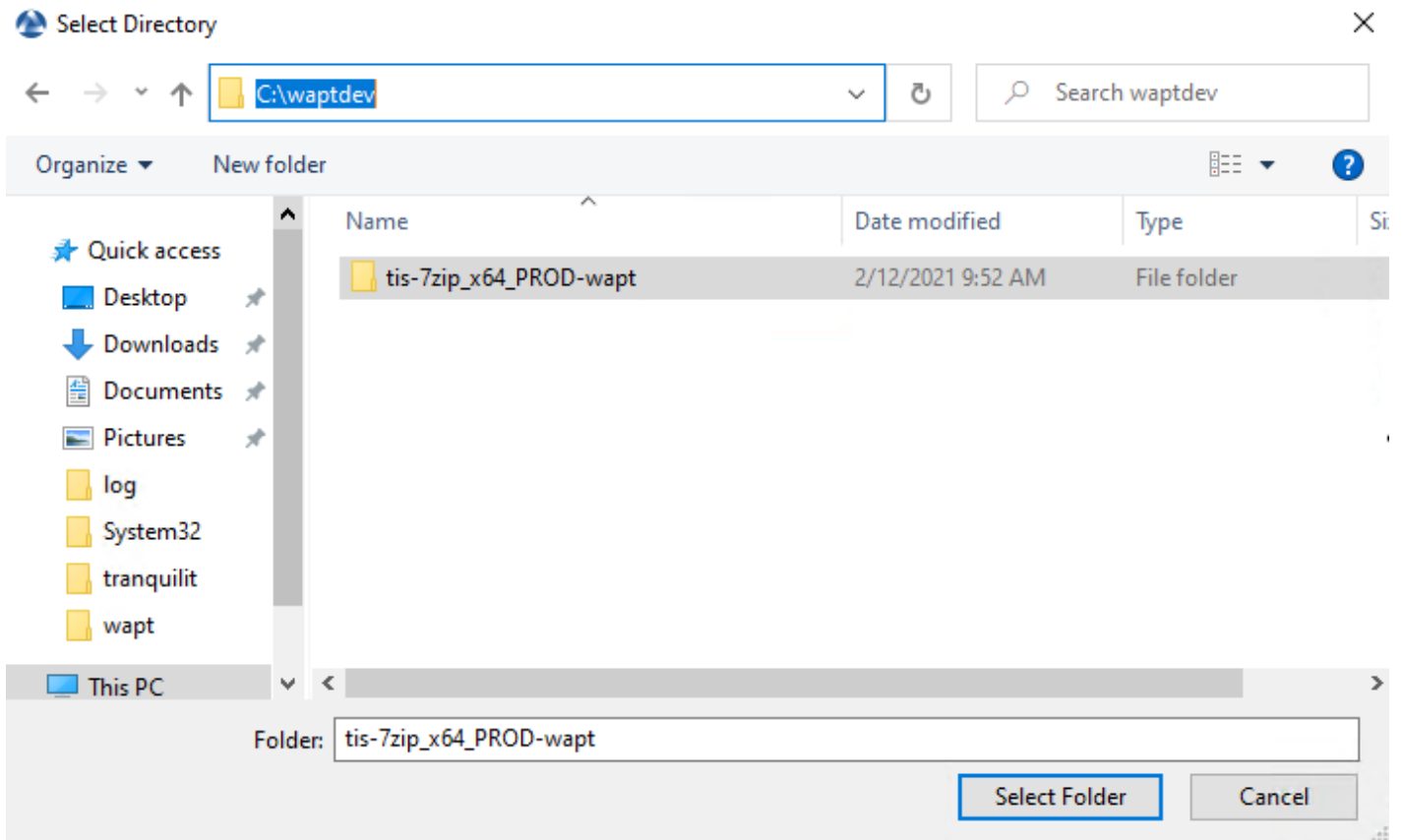
- Sélectionner le paquet dans le dossier `c:\waptdev`.
- Confirmer le paquet sélectionné.

Vous venez de charger votre premier paquet wapt.

---

**Note :** Une ancienne méthode en ligne de commande est disponible ici.

---



**Avertissement :** Une fois que votre paquet est téléversé, rafraîchissez la liste des paquets en utilisant le bouton *Actualiser les paquets disponibles* ou en appuyant sur la touche F5 de votre clavier.

## 50.1 Travailler avec des codes de retour non standard

Les codes de retour sont utilisés pour indiquer si un logiciel a été correctement installé.

Avec Windows, le code standard de retour pour une installation réussie est [0].

Si vous savez que vos paquets WAPT s'installent correctement, mais que vous obtenez quand même un code de retour différent de [0], alors vous pouvez explicitement dire à WAPT d'ignorer le code d'erreur en utilisant le paramètre `accept_returncodes`.

Vous pouvez découvrir comment utiliser le paramètre `accept_returncodes` en explorant le code de ce paquet.

```
# -*- coding: utf-8 -*-
from setuphelpers import *
import re

uninstallkey = []

def is_kb_installed(hotfixid):
    installed_update = installed_windows_updates()
    if [kb for kb in installed_update if kb['HotFixID' ].upper() == hotfixid.upper()]:
        return True
    return False

def waiting_for_reboot():
    # Query WUAU from the registry
    if reg_key_exists(HKEY_LOCAL_MACHINE, r"SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\
↪Auto Update\RebootRequired") or \
        reg_key_exists(HKEY_LOCAL_MACHINE, r"SOFTWARE\Microsoft\Windows\CurrentVersion\Component_
↪Based Servicing\RebootPending") or \
        reg_key_exists(HKEY_LOCAL_MACHINE, r'SOFTWARE\Microsoft\Updates\UpdateExeVolatile'):
        return True
    return False

def install():
    kb_files = [
        'windows10.0-kb4522355-x64_af588d16a8fbb572b70c3b3bb34edee42d6a460b.msu',
    ]
    with EnsureWUAUServRunning():
        for kb_file in kb_files:
            kb_guess = re.findall(r'^.*-(KB.*)-', kb_file)
            if not kb_guess or not is_kb_installed(kb_guess[0]):
                print('Installing {}'.format(kb_file))
                run('wusa.exe "{}" /quiet /norestart'.format(kb_file), accept_returncodes=[0, 3010,
↪2359302, -2145124329], timeout=3600)
            else:
                print('{} already installed'.format(kb_file))
```

(suite sur la page suivante)

(suite de la page précédente)

```
if waiting_for_reboot():  
    print('A reboot is needed!')
```

---

**Indication :** La liste complète des messages d'erreur de l'installeur Windows peut être consultée sur cette page <<https://docs.microsoft.com/en-us/windows/win32/msi/windows-installer-error-messages>>`\_.

---

---

## Exemples simples de fonctions du setuphelper couramment utilisées

---

Nous présentons ici quelques fonctions implémentées dans *Setuphelpers* et fréquemment utilisées pour développer des paquets WAPT.

### 51.1 Tests et manipulation de dossiers et fichiers

#### 51.1.1 Créer un chemin avec récursion

La commande **makepath** ...

```
makepath(programfiles, 'Mozilla', 'Firefox')
```

... fabrique la variable pour le chemin C:\Program Files (x86)\Mozilla\Firefox.

---

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

<https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=makepath#setuphelpers.makepath>

---

#### 51.1.2 Créer et supprimer des raccourcis

La commande **mkdirs** ...

```
mkdirs('C:\\test')
```

... crée le répertoire C:\test.

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

<https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=mkdirs#setuphelpers.mkdirs>

---

La commande **remove\_tree** ...

```
remove_tree(r'C:\tmp\target')
```

... détruit le répertoire C:\tmp\target.

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=remove\\_tree#setuphelpers.remove\\_tree](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=remove_tree#setuphelpers.remove_tree)

---

### 51.1.3 Vérifier si un chemin est un fichier ou un dossier

La commande **isdir** ...

```
isdir(makepath(programfiles32, 'software')):  
    print('The directory exists')
```

... vérifie que C:\Program Files (x86)\software est un répertoire.

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

<https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=isdir#setuphelpers.isdir>

---

La commande **isfile** ...

```
isfile(makepath(programfiles32, 'software', 'file')):  
    print('file exist')
```

... vérifie que C:\Program Files (x86)\software\file est un fichier.

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

<https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=isfile#setuphelpers.isfile>

---



### 51.1.4 Vérifier si un répertoire est vide

La commande `dir_is_empty` ...

```
dir_is_empty(makepath(programfiles32, 'software')):  
    print('dir is empty')
```

... vérifie que le répertoire C:\Program Files (x86)\software est vide.

---

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=dir\\_is\\_empty#setuphelpers.dir\\_is\\_empty](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=dir_is_empty#setuphelpers.dir_is_empty)

---

### 51.1.5 Copier un fichier

La commande `filecopyto` ...

```
filecopyto('file.txt', makepath(programfiles32, 'software'))
```

... copie le fichier `file.txt` dans le répertoire C:\Program Files (x86)\software.

---

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

<https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=filecopyto#setuphelpers.filecopyto>

---

### 51.1.6 Copier un dossier

La commande `copytree2` ...

```
copytree2('sources', 'C:\\projet')
```

... copie le dossier `sources` dans le répertoire C:\projet.

---

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

<https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=copytree2#setuphelpers.copytree2>

---

### 51.1.7 Récupérer la version d'un fichier

La commande `get_file_properties` ...

```
get_file_properties(makepath(programfiles32, 'InfraRecorder', 'infrarecorder.exe'))['ProductVersion']
```

... affiche les propriétés du paquet.

---

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=get\\_file\\_properties#setuphelpers.get\\_file\\_properties](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=get_file_properties#setuphelpers.get_file_properties)

---

---

## Manipulation de clés de registre

---

### 52.1 Vérifier l'existence d'une clé de registre

La commande `registry_readstring` ...

```
if registry_readstring(HKEY_LOCAL_MACHINE, "SOFTWARE\\Google\\Update\\Clients\\{8A69D345-D564-463c-  
↪AFF1-A69D9E530F96}", 'pv'):  
    print('key exist')
```

... vérifie que la clé `{8A69D345-D564-463c-AFF1-A69D9E530F96}` existe dans le répertoire `SOFTWARE\Google\Update\Clients` de la ruche `HKEY_LOCAL_MACHINE`.

---

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=registry\\_readstring#setuphelpers.registry\\_readstring](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=registry_readstring#setuphelpers.registry_readstring)

---

### 52.2 Afficher la valeur d'une clé de registre

La commande `registry_readstring` ...

```
print(registry_readstring(HKEY_LOCAL_MACHINE, r'SOFTWARE\Google\Update\Clients\{8A69D345-D564-463c-  
↪AFF1-A69D9E530F96}', 'pv'))
```

... lit la valeur de la clé `{8A69D345-D564-463c-AFF1-A69D9E530F96}` inscrite dans le répertoire `SOFTWARE\Google\Update\Clients` de la ruche `HKEY_LOCAL_MACHINE`.

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=registry\\_readstring#setuphelpers.registry\\_readstring](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=registry_readstring#setuphelpers.registry_readstring)

---

## 52.3 Modifier la valeur d'une clé de registre

La commande `registry_setstring` ...

```
registry_setstring(HKEY_CURRENT_USER, "SOFTWARE\\Microsoft\\Windows Live\\Common", 'TOUVersion', '16.  
↪0.0.0', type=REG_SZ)
```

... modifie la valeur de la clé `TOUVersion` dans le répertoire `SOFTWARE\\Microsoft\\Windows Live` de la ruche `HKEY_CURRENT_USER`.

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=registry\\_setstring#setuphelpers.registry\\_setstring](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=registry_setstring#setuphelpers.registry_setstring)

---

---

## Créer et supprimer des raccourcis

---

With WAPT setuphelper it's possible to create different type of shortcut.

### 53.1 Un raccourci simple

---

À faire : upcoming documentation

---

### 53.2 Un raccourci dans le menu démarrer

---

À faire : upcoming documentation

---

### 53.3 Créer et supprimer des raccourcis

---

À faire : upcoming documentation

---

### 53.3.1 créer un raccourci personnalisé sur le bureau de l'utilisateur en cours

La commande `create_desktop_shortcut` ...

```
create_desktop_shortcut(r'WAPT Console Management', target=r'C:\Program Files (x86)\wapt\waptconsole.exe')
```

... crée le raccourci *WAPT Console Management* dans le répertoire `C:\Users\Public` pointant vers l'exécutable `C:\Program Files (x86)\wapt\waptconsole.exe`; le raccourci est ajouté pour tous les utilisateurs.

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=create\\_desktop\\_shortcut#setuphelpers.create\\_desktop\\_shortcut](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=create_desktop_shortcut#setuphelpers.create_desktop_shortcut)

---

### 53.3.2 supprimer des raccourcis

La commande `remove_desktop_shortcut` ...

```
remove_desktop_shortcut('WAPT Console Management')
```

... supprime le raccourci *WAPT Console Management* du répertoire `C:\Users\Public`; le raccourci est supprimé pour tous les utilisateurs.

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=remove\\_desktop\\_shortcut#setuphelpers.remove\\_desktop\\_shortcut](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=remove_desktop_shortcut#setuphelpers.remove_desktop_shortcut)

Firefox place un raccourci sur le bureau de tous les utilisateurs, nous allons le supprimer.

---

Nous utiliserons la fonction `remove_desktop_shortcut` :

- Modifier le `setup.py` et utiliser la fonction comme ceci.

```
# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = []

def install():
    install_exe_if_needed("Firefox Setup 45.5.0esr.exe", silentflags="-ms", key='Mozilla Firefox',
↳45.5.0 ESR (x64 fr)', min_version="45.5.0")
    remove_desktop_shortcut('Firefox')
```

- Si vous redémarrez l'installation à partir de **pyscripter**, vous remarquerez que le raccourci de bureau « all users » a disparu.

## 53.4 Créer un raccourci personnalisé sur le bureau de l'utilisateur en cours

### 53.4.1 créer un raccourci personnalisé sur le bureau de l'utilisateur en cours

---

**Indication :** Ces fonctions sont utilisées avec le `session_setup`.

---

La commande `create_user_desktop_shortcut` ...

```
create_user_desktop_shortcut(r'WAPT Console Management',target=r'C:\Program Files (x86)\wapt\  
↪waptconsole.exe')
```

... crée le raccourci *WAPT Console Management* sur le bureau de l'utilisateur pointant vers l'exécutable **C:\Program Files (x86)\wapt\waptconsole.exe**.

---

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=create\\_user\\_desktop\\_shortcut#setuphelpers.create\\_user\\_desktop\\_shortcut](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=create_user_desktop_shortcut#setuphelpers.create_user_desktop_shortcut)

---

### 53.4.2 supprimer un raccourci personnalisé sur le bureau de l'utilisateur en cours

La commande `remove_user_desktop_shortcut` ...

```
remove_user_desktop_shortcut('WAPT Console Management')
```

... supprime le raccourci *WAPT Console Management* du bureau de l'utilisateur.

---

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=remove\\_user\\_desktop\\_shortcut#setuphelpers.remove\\_user\\_desktop\\_shortcut](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=remove_user_desktop_shortcut#setuphelpers.remove_user_desktop_shortcut)

---





## 54.1 Vérifier la version de Windows

La commande `windows_version` ...

```
windows_version(<Version('6.2.0')>:
```

... vérifie que la version de Windows est strictement inférieure à 6.2.0.

---

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuptools.html?highlight=windows\\_version#setuptools.windows\\_version](https://www.wapt.fr/en/api-doc-1.5/source/setuptools.html?highlight=windows_version#setuptools.windows_version)

Visitez également le site sur les numéros de version de Windows.

---

## 54.2 Vérifier si l'architecture est en 64bits

La commande `iswin64` ...

```
if iswin64():  
    print('Pc x64')  
else:  
    print('Pc not x64')
```

... vérifie que le processeur de la machine est 64bits.

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

<https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=iswin64#setuphelpers.iswin64>

---

## 54.3 La variable Program Files

programfiles / programfiles32 / programfiles64

Renvoie les différentes localisations de *Program Files*

La commande **programfiles64** ...

```
print(programfiles64())
```

Renvoie le répertoire natif du programme, c.à.d. C:\Program Files (x86) pour les architecture win64 et win32.

```
print(programfiles())
```

Renvoie le chemin vers le répertoire Programs Files (x86) (sur architecture win64) ou Programs Files (sur architecture win32).

```
print(programfiles32())
```

## 54.4 La variable AppData

user\_appdata / user\_local\_appdata

---

**Indication :** Ces fonctions sont utilisées avec le **session\_setup**

---

La commande **user\_appdata** ...

```
print(user_appdata())
```

... renvoie le profil *appdata* itinérant de l'utilisateur courant (C:\Users\%username%\AppData\Roaming).

---

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=user\\_appdata#setuphelpers.user\\_appdata](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=user_appdata#setuphelpers.user_appdata)

---

La commande **user\_local\_appdata** ...

```
print(user_local_appdata())
```

... renvoie le profil *appdata* local de l'utilisateur courant (C:\Users\%username%\AppData\Local).

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=user\\_local\\_appdata#setuphelpers.user\\_local\\_appdata](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=user_local_appdata#setuphelpers.user_local_appdata)

---

## 54.5 Désactiver temporairement le redirecteur wow3264

La commande `disable_file_system_redirection` ...

```
with disable_file_system_redirection():  
    filecopyto('file.txt', system32())
```

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=disable\\_file\\_system\\_redirection#setuphelpers.disable\\_file\\_system\\_redirection](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=disable_file_system_redirection#setuphelpers.disable_file_system_redirection)

---

Gère le contexte pour désactiver temporairement le redirecteur wow3264

## 54.6 Récupérer l'utilisateur courant

La commande `get_current_user` ...

```
print(get_current_user())
```

... affiche l'identifiant de l'utilisateur connecté

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=get\\_current\\_user#setuphelpers.get\\_current\\_user](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=get_current_user#setuphelpers.get_current_user)

---

## 54.7 Récupérer le nom de l'ordinateur

La commande `get_computername` ...

```
print(get_computername())
```

... affiche le nom de la machine sans le domaine

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=get\\_computername#setuphelpers.get\\_computername](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=get_computername#setuphelpers.get_computername)

---

## 54.8 Récupérer le nom de domaine

La commande `get_domain_fromregistry` ...

```
get_domain_fromregistry()
```

... renvoie le nom de la machine avec le domaine.

---

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=get\\_domain\\_fromregistry#setuphelpers.get\\_domain\\_fromregistry](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=get_domain_fromregistry#setuphelpers.get_domain_fromregistry)

---

## 54.9 Les actions sur les logiciels installés

### 54.9.1 Vérifier les logiciels installés

La commande `installed_softwares` ...

```
installed_softwares('winscp')
```

... renvoie la liste des logiciels inscrits dans la base de registre machine sous forme de tableau.

```
[{'install_location': u'C:\\Program Files\\WinSCP\\', 'version': u'5.9.2', 'name': u'WinSCP 5.9.2',  
↪ 'key': u'winscp3_is1', 'uninstall_string': u'"C:\\Program Files\\WinSCP\\unins000.exe"',  
↪ 'publisher': u'Martin Prikryl', 'install_date': u'20161102', 'system_component': 0}]
```

---

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=installed\\_softwares#setuphelpers.installed\\_softwares](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=installed_softwares#setuphelpers.installed_softwares)

---

### 54.9.2 Récupérer la commande de désinstallation avec le registre

La commande `uninstall_cmd` ...

```
uninstall_cmd('winscp3_is1')
```

... renvoie la commande de désinstallation silencieuse.

```
"C:\\Program Files\\WinSCP\\unins000.exe" /SILENT
```

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=uninstall\\_cmd#setuphelpers.uninstall\\_cmd](https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=uninstall_cmd#setuphelpers.uninstall_cmd)

---

### 54.9.3 Désinstaller des logiciels

```
for soft in installed_softwares('winscp3'):
    if Version(soft['version']) < Version('5.0.2'):
        run(WAPT.uninstall_cmd(soft['key']))
```

- Pour chaque élément de la liste renvoyée par *installed\_softwares* contenant le mot-clé *winscp*.
- Si la version dans la liste est plus petite que 5.0.2.
- Alors lancer la désinstallation avec *uninstall\_cmd* et en indiquant la *uninstallkey*.

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

[https://dev.tranquil.it/sphinxdocs/source/setuphelpers.html?highlight=uninstall\\_cmd#setuphelpers.uninstall\\_cmd](https://dev.tranquil.it/sphinxdocs/source/setuphelpers.html?highlight=uninstall_cmd#setuphelpers.uninstall_cmd)

---

### 54.9.4 Tuer des tâches

La commande **killalltasks** ...

```
killalltasks('firefox')
```

... termine l'exécution du logiciel *Firefox*.

**Indication :** Pour plus d'informations ou pour connaître les paramètres complémentaires de la commande, consultez la documentation de référence (en anglais) en visitant :

<https://www.wapt.fr/en/api-doc-1.5/source/setuphelpers.html?highlight=killalltasks#setuphelpers.killalltasks>

---



---

## Utiliser les champs du fichier control

---

Il est possible d'utiliser les informations du fichier control dans le `setup.py`

### 55.1 Récupérer la version du paquet

```
def setup():  
    print(control['version'])
```

... affiche le champ `version` du fichier `control` du paquet WAPT.

```
def setup():  
    print(control['version'].split('-',1)[0])
```

... affiche le numéro de version du fichier `control` sans le numéro de version de packaging WAPT.

### 55.2 Récupérer le nom du logiciel

---

**À faire :** upcoming documentation

---





---

## Gérer un paquet WAPT avec un autre paquet WAPT

---

### 56.1 Installer un paquet

La commande **install** ...

```
WAPT.install('tis-scratch')
```

... installe *tis-scratch* sur la machine.

### 56.2 Supprimer un paquet

La commande **remove** ...

```
WAPT.remove('tis-scratch')
```

... désinstalle *tis-scratch* de la machine.

### 56.3 Créer des paquets WAPT

La commande **forget\_packages** ...

```
WAPT.forget_packages('tis-scratch')
```

... informe WAPT de ne plus suivre le paquet *tis-scratch*; WAPT ne connaîtra plus l'existence de ce paquet.

---

**Indication :** Si vous voulez supprimer *tis-scratch*, il faudra soit réinstaller le paquet (**wapt-get install "tis-scratch"**), puis le supprimer (**wapt-get remove "tis-scratch"**), ou bien le supprimer manuellement à partir du panneau de configuration Windows *Ajout / Suppression de Programmes*.

---

## 57.1 Copier un fichier

Il est possible de configurer **Firefox** avec un fichier `policies.json`. Voir <https://github.com/mozilla/policy-templates/blob/master/README.md>.

Ce fichier doit être placé dans le dossier `distribution` à la racine de Firefox.

Pour vous aider à créer ce fichier `policies.json`, vous pouvez utiliser cette extension : <https://addons.mozilla.org/fr/firefox/addon/enterprise-policy-generator/>.

Lorsque vous avez généré votre fichier `policies.json`, placez-le dans `c:\waptdev\prefix-firefox-esr-wapt\policies.json`.

Le dossier `distribution` à la racine de Firefox peut ne pas exister, nous allons donc tester son existence et le créer avec la commande `mkdirs` si il n'existe pas :

```
if not isdir(r'C:\Program Files\Mozilla Firefox\distribution'):
    mkdirs(r'C:\Program Files\Mozilla Firefox\distribution')
```

---

**Important :** Si vous avez des *backslashes* sur votre chemin, vous devez toujours mettre un `r` devant la chaîne, comme dans l'exemple précédent.

Vous devrez également utiliser la fonction `filecopyto` pour copier le fichier `policies.json` :

```
filecopyto('policies.json',r'C:\Program Files\Mozilla Firefox\distribution')
```

---

**Indication :** Il n'est pas nécessaire de mettre le chemin complet du fichier source puisque le fichier `policies.json` est à la racine du paquet WAPT, donc nous utilisons le chemin relatif.

---

Modifier le `setup.py` :

```
# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = []

def install():
    install_exe_if_needed("Firefox Setup 45.5.0esr.exe", silentflags="-ms", key='Mozilla Firefox 45.5.
→0 ESR (x64 fr)', min_version="45.5.0")
    remove_desktop_shortcut('Firefox')

    if not isdir(r'C:\Program Files\Mozilla Firefox\distribution'):
        mkdirs(r'C:\Program Files\Mozilla Firefox\distribution')

    filecopyto('policies.json', r'C:\Program Files\Mozilla Firefox\distribution')
```

Votre paquet est maintenant prêt à appliquer une configuration. Vous pouvez lancer une installation avec **pyscripter** et valider que le paquet fonctionne selon votre objectif.

Enfin, lancer votre **Firefox** pour vérifier qu'il fonctionnera pour vos utilisateurs.

## 57.2 Désinstaller des versions non désirées

---

**Indication** : À chaque étape de ces exemples, vous pouvez lancer une installation pour tester le résultat.

---

Dans notre cas, nous voulons désinstaller la version non ESR de **Firefox**.

Nous chercherons les autres logiciels installés sur la machine pour vérifier si une version non-esr de **Firefox** est installée.

Pour reproduire notre exemple, téléchargez et installez la version grand public ici : <https://download.mozilla.org/?product=firefox-latest-ssl&os=win> :

- Pour rechercher une version non désirée de **Firefox**, nous utiliserons la fonction `installed_softwares`. Cette fonction renvoie un dictionnaire contenant les propriétés du logiciel :

```
print(installed_softwares('Firefox'))

[
  {
    'install_date': '',
    'install_location': 'C:\\Program Files\\Mozilla Firefox',
    'key': 'Mozilla Firefox 78.7.1 ESR (x64 fr)',
    'name': 'Mozilla Firefox 78.7.1 ESR (x64 fr)',
    'publisher': 'Mozilla',
    'system_component': 0,
    'uninstall_string': '"C:\\Program Files\\Mozilla Firefox\\uninstall\\helper.exe"',
    'version': '78.7.1',
    'win64': True
  },
]
```

(suite sur la page suivante)

(suite de la page précédente)

```

{
'install_date': '',
'install_location': 'C:\Program Files (x86)\Mozilla Firefox',
'key': 'Mozilla Firefox 79.0 (x86 fr)',
'name': 'Mozilla Firefox 79.0 (x86 fr)',
'publisher': 'Mozilla',
'system_component': 0,
'uninstall_string': '"C:\Program Files (x86)\Mozilla Firefox\uninstall\helper.exe"',
'version': '79.0',
'win64': False
}
]

```

— Vérifier le nom de chaque logiciel.

```

for uninstall in installed_softwares('Mozilla Firefox'):
    print(uninstall['name'])

```

— Afficher le nom de chaque logiciel trouvé.

```

for uninstall in installed_softwares('Mozilla Firefox'):
    if not 'ESR' in uninstall['name']:
        print(uninstall['name'])

```

— Afficher le nom de chaque logiciel trouvé qui n'inclut pas la chaîne *ESR* dans son nom et sa clé de désinstallation.

```

for uninstall in installed_softwares('Mozilla Firefox'):
    if not 'ESR' in uninstall['name']:
        print(uninstall['name'])
        print('Uninstall ' + uninstall['key'])

```

Nous allons maintenant utiliser une astuce WAPT en utilisant la fonction `uninstall_cmd` :

— `Install cmd` accepte une clé de désinstallation comme argument et enverra la commande à exécuter pour lancer la désinstallation silencieuse.

```

for uninstall in installed_softwares('Mozilla Firefox'):
    if not 'ESR' in uninstall['name']:
        print(uninstall['name'])
        print('Uninstall ' + uninstall['key'])
        silent_uninstall = uninstall_cmd(uninstall['key'])
        print('Run ' + silent_uninstall)

```

— Commencer la désinstallation.

```

for uninstall in installed_softwares('Mozilla Firefox'):
    if not 'ESR' in uninstall['name']:
        print(uninstall['name'])
        print('Uninstall ' + uninstall['key'])
        silent_uninstall = uninstall_cmd(uninstall['key'])
        print('Run ' + silent_uninstall)
        run(silent_uninstall)

```

Nous pouvons également désinstaller le service de maintenance de mozilla :

```
for uninstall in installed_softwares('MozillaMaintenanceService'):
    run(uninstall_cmd(uninstall['key']))
```

— Enfin, modifier votre `setup.py` :

```
# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = []

def install():
    #Install firefox if necessary
    install_exe_if_needed("Firefox Setup 45.5.0esr.exe",silentflags="-ms",key='Mozilla Firefox',
↳45.5.0 ESR (x64 fr)',min_version="45.5.0")

    #Removal of the firefox shortcut on the all user desktop
    remove_desktop_shortcut('Firefox')

    #Creation of the distribution folder if it does not exist
    if not isdir(r'C:\Program Files\Mozilla Firefox\distribution'):
        mkdirs(r'C:\Program Files\Mozilla Firefox\distribution')

    #Copy of the policies.json file found at the root of the package in the destination of the
↳distribution folder
    filecopyto('policies.json',r'C:\Program Files\Mozilla Firefox\distribution')

    #For each Mozilla Firefox installed
    for uninstall in installed_softwares('Mozilla Firefox'):
        #If the software does not have the word ESR in the name
        if not 'ESR' in uninstall['name']:
            print(uninstall['name'])
            print('Uninstall ' + uninstall['key'])

            #Looking for how we can uninstall it silently
            silent_uninstall = uninstall_cmd(uninstall['key'])
            print('Run ' + silent_uninstall)

            #We launch the previous command.
            run(silent_uninstall)

    #Uninstalling mozilla maintenance service
    for uninstall in installed_softwares('MozillaMaintenanceService'):
        run(uninstall_cmd(uninstall['key']))
```

Votre code gère maintenant la désinstallation des versions non désirées de **Firefox**.

## 57.3 Améliorer setup.py pour utiliser des variables

Exemples d'utilisation de variables :

```
version_firefox = "45.0"

uninstallkey = "Mozilla Firefox " + version_firefox + " ESR (x64 fr)"
print(uninstallkey)

uninstallkey = "Mozilla Firefox %s ESR (x64 fr)" % (version_firefox)
print(uninstallkey)

uninstallkey = "Mozilla Firefox {} ESR (x64 fr)".format(version_firefox)
print(uninstallkey)

uninstallkey = f"Mozilla Firefox {version_firefox} ESR (x64 fr)"
print(uninstallkey)
```

**Important :** Le dernier exemple est le meilleur mais cette opération ne fonctionne qu'avec **Python3**.

Nous pouvons maintenant utiliser des variables dans notre fichier setup.py :

```
# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = []

def install():

    version_firefox = "45.5.0"

    #Install firefox if necessary
    install_exe_if_needed("Firefox Setup %sesr.exe" % version_firefox, silentflags="-ms",
    ↪key='Mozilla Firefox %s ESR (x64 fr)' % version_firefox, min_version=version_firefox)

    #Removal of the firefox shortcut on the all user desktop
    remove_desktop_shortcut('Firefox')

    distribution_folder=r'C:\Program Files\Mozilla Firefox\distribution'

    #Creation of the distribution folder if it does not exist
    if not isdir(distribution_folder):
        mkdirs(distribution_folder)

    ... The rest of the code does not change ...
```

**Indication :** Vous pouvez récupérer le numéro de version indiqué dans le fichier control comme ceci :

```
version_firefox = control.get_software_version()
```

---

## 57.4 Personnaliser le contexte utilisateur

Il est parfois nécessaire de personnaliser un programme ou un logiciel en contexte utilisateur pour rendre le logiciel immédiatement exploitable par l'utilisateur dans le contexte spécifique de son entreprise ou du service au sein de son entreprise :

- Créer des raccourcis sur le bureau utilisateur avec des arguments spécifiques.
- Modifier de clés registres utilisateurs.
- Modifier des fichiers, une configuration de navigateur.
- Configurer des raccourcis réseaux aux modèles de documents de l'entreprise pour assurer la conformité des documents aux chartes éditoriales en vigueur.
- Paramétrer la messagerie instantanée ou le mail de l'utilisateur à partir d'un référentiel externe (annuaire, base de données, etc) .
- Paramétrer un logiciel bureautique ou métier à partir d'un référentiel externe (annuaire, base de données, etc).

La fonction **session\_setup** bénéficie de toute la puissance et de l'étendue des librairies python pour atteindre un niveau d'automatisation élevé.

### 57.4.1 Les principes du *session\_setup*

La fonction WAPT : command *session\_setup* est exécutée pour chaque utilisateur utilisant cette commande :

```
C:\Program Files (x86)\wapt\wapt-get.exe session-setup ALL
```

L'appel à cette fonction permet d'exécuter la partie **session\_setup** de chaque paquet WAPT logiciel installé sur la machine.

WAPT enregistre en base locale les instructions de tous les paquets dans le fichier C:\Program Files (x86)\wapt\waptdb.sqlite.

**Attention :** Le **session\_setup** de chaque paquet n'est exécuté qu'**une seule fois par paquet ou version de paquet et par profil utilisateur**.

L'agent WAPT stocke dans la base de données locale %appdata%\wapt\waptsession.sqlite les instances de **session\_setup** qui ont déjà été jouées.

Exemple de sortie de la commande `wapt-get session-setup ALL` :

---

**Note :** le *session\_setup* de l'utilisateur connecté, avait déjà été exécuté.

---

```
wapt-get session-setup ALL

Configuring tis-7zip ... No session-setup. Done
Configuring tis-ccleaner ... Already installed. Done
Configuring tis-vlc ... No session-setup. Done
Configuring tis-tightvnc ... No session-setup. Done
```

(suite sur la page suivante)



(suite de la page précédente)

```
Configuring tis-paint.net ... No session-setup. Done
Configuring wsuser01.mydomain.lan ... No session-setup. Done
```

## 57.4.2 Utiliser le session-setup

Les scripts `session_setup` sont situés dans la section `def session_setup()` du fichier `setup.py` :

Exemple :

```
def session_setup():
    registry_setstring(HKEY_CURRENT_USER, "SOFTWARE\\Microsoft\\Windows Live\\Common", 'TOUVersion',
↳ '16.0.0.0', type=REG_SZ)
```

**Attention :** Avec `session_setup`, il n'est pas possible de faire appel à des fichiers contenus dans le paquet.

Pour appeler des fichiers externes lors de la désinstallation, copier et coller les fichiers nécessaires dans un dossier externe pendant le processus d'installation du paquet (exemple : `c:cachefile`).

## 57.4.3 Exemple : Créer un raccourci personnalisé sur le bureau

Une des possibilités offertes par *Setuphelpers* est la création de raccourcis individuels sur le bureau utilisateur, à la différence du bureau « Public » commun à tous les utilisateurs.

Nous utiliserons pour ça la fonction `create_user_desktop_shortcut()` pour créer un raccourci contenant le nom de l'utilisateur et qui passera en argument à Firefox le site <https://tranquil.it> par exemple.

```
# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = []

def install():
    install_exe_if_needed("Firefox Setup 45.5.0esr.exe", silentflags="-ms", key='Mozilla Firefox 45.4.
↳ ESR (x64 fr)', min_version="45.5.0")

def session_setup():
    create_user_desktop_shortcut("Mozilla Firefox de %s" % get_current_user(), r'C:\Program Files\
↳ Mozilla Firefox\firefox.exe', arguments="-url https://tranquil.it")
```

- Maintenant, lancer le `session-setup` directement à partir de **pyscripter**.
- Enfin, vérifier que l'icône est bien présente sur le bureau.

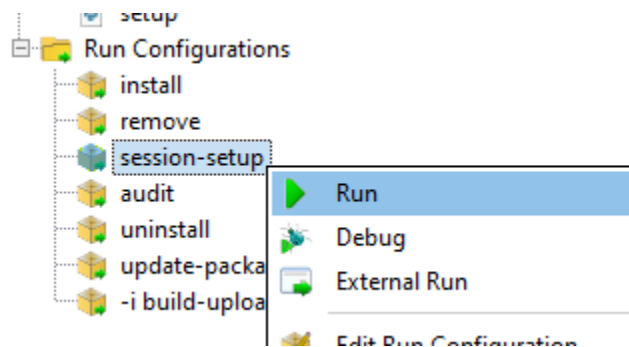


FIG. 1 – Pyscripter - Effectuer un session-setup

## 57.5 Utiliser les fonctions d'audit pour la conformité

**Note :** Cette fonctionnalité est disponible dans la version **Entreprise**.

L'audit permet d'effectuer des vérifications régulières sur les configurations des postes et de centraliser le résultat des vérifications dans la console WAPT. Ceci permet de vérifier que votre parc est conforme à votre référentiel sur la durée.

Vous pouvez par exemple :

- Vérifier régulièrement la liste des administrateurs locaux des postes.
- Vérifier régulièrement la bonne configuration d'un logiciel critique.
- Vérifier régulièrement la présence de la bonne version d'un logiciel.
- Vérifier régulièrement les configurations de sécurité d'un poste.

La fonction **audit** bénéficie de toute la puissance et de l'étendue des bibliothèques python pour atteindre une précision d'audit élevée.

### 57.5.1 Principe de fonctionnement

Les tâches d'**audit** s'exécutent après un **upgrade** puis à intervalle régulier défini par la valeur de `audit_schedule`.

Pour exécuter manuellement un audit vous pouvez également exécuter la commande :

```
wapt-get audit
```

**Note :** Par défaut, la fonction `audit` ne sera pas lancée si l'audit n'est pas nécessaire.

Pour forcer l'exécution vous pouvez exécuter la commande :

```
wapt-get audit -f
```

On définit la partie **audit** dans le fichier `setup.py` du paquet dans une fonction `def audit()` :

Dans cet exemple, nous améliorons le paquet `firefox` précédemment étudié dans cette documentation.

- Ajouter la fonction `audit` dans le fichier `setup.py`.

```
def audit():
    if isfile(r'C:\Program Files\Mozilla Firefox\distribution\policies.json'):
        print('File policies.json found')
        return "OK"
    else:
        print('File policies.json not found')
        return "ERROR"
```

— Lancer l’audit à partir de **pyscripter**.

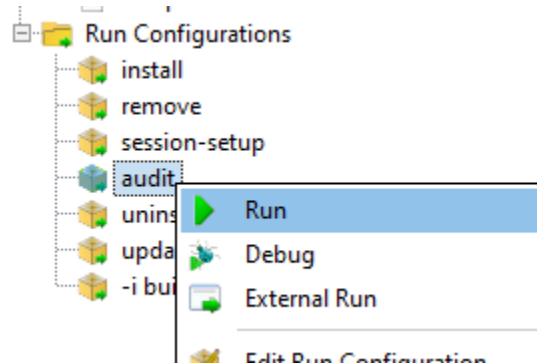


FIG. 2 – Pyscripter - Effectuer un audit

— Tester avec le fichier puis supprimer le fichier C:\Program Files\Mozilla Firefox\distribution\policies.json et tester à nouveau avec **pyscripter**.

Vous pouvez voir directement l’état de l’audit dans la console (Cliquez sur le paquet puis sur la colonne audit) :

L’audit restitue une des 3 valeurs suivantes :

- **OK** ;
- **WARNING** ;
- **ERROR** ;

**Attention** : Avec **audit**, il n’est pas possible de faire appel à des fichiers contenus dans le paquet.

Pour utiliser des fichiers lors de l’audit il faut d’abord les copier dans un répertoire temporaire de la machine lors de l’installation du paquet.

### 57.5.2 Planifier un audit

Les tâches d’**audit** s’exécutent après un **upgrade** puis à intervalle régulier défini par la valeur de **audit\_schedule**.

La valeur est contenue dans le fichier **control** du paquet WAPT.

Par défaut si **audit\_schedule** est vide alors il faudra effectuer l’audit manuellement ou à partir de la console WAPT.

Sinon la valeur peut être indiquée de plusieurs manières :

- Un entier (en minutes).
- Un entier suivi d’une lettre (m = minutes , h = heure , d = jour , w = semaine).

Status	Reachable	Audit status	WUA	Host
T...	OK	ERROR	OK	laptop-t430.ad.tranquil.it

Overview Hardware inventory Software inventory Windows u

Name: LAPTOP-T430

Description:

Operating system: Windows 10 Pro

IP address: 192.168.1.48,192.168.56.1,172.16.144.52

Last task:

Search keywords   Errors  To upgrade

Statu	Audit status	Package name	Versic
	OK	OU=laptops_OU=computers_O...	19
	OK	base	201
	ERROR	demo-firefox-esr	80-2
	OK	grp-laptop	10
	OK	grp-security	16
	OK	tis-3cxphone-for-tis	4.0.2
	OK	tis-7zip	19.0-

Selected /

**Audit logs of package demo-firefox-esr**

File policies.json not found

FIG. 3 – Vérifier l'état d'un audit dans la console WAPT

### 57.5.3 Comportement par défaut de la fonction d'audit

Par défaut, si aucun audit n'est déclaré, l'agent WAPT vérifiera la présence des *Uninstallkey* dans le paquet WAPT.

De cette manière WAPT vérifie que le logiciel est toujours présent.

## 57.6 Automatiser la mise à jour d'un paquet logiciel

---

**Note :** Cette partie de la documentation est déconseillée aux utilisateurs qui débutent avec WAPT.

---

Les fonctions *update\_package* sont très pratiques, elles permettent de gagner du temps lorsque qu'il faut mettre à jour un paquet avec la version la plus récente d'un logiciel.

### 57.6.1 Principe de fonctionnement

La fonction *update\_package* paquet ira :

- Récupérer la dernière version du logiciel en ligne.
- Télécharger la dernière version du binaire.
- Supprimer les anciennes version des binaires.
- Mettre à jour le numéro de version dans le fichier *control*.

Si votre fonction *install* se base sur la version du fichier *control* pour l'installation, alors vous n'avez pas besoin de modifier votre *setup.py*.

Il vous reste maintenant à tester l'installation avant de lancer un **build-upload**.

### 57.6.2 Exemple

Voici l'*update\_package* de **firefox-esr** comme exemple :

```
def update_package():
    import re, requests, glob

    #Retrieving the last file name
    url = requests.head('https://download.mozilla.org/?product=firefox-esr-latest&os=win64',
↳proxies={}).headers['Location']
    filename = url.rsplit('/',1)[1].replace('%20', ' ')

    #download of it if is not in the package
    if not isfile(filename):
        print('Downloading %s from %s'%(filename,url))
        wget(url, filename)

    #removing old exe with wrong name
    for fn in glob.glob('*.exe'):
        if fn != filename:
            remove_file(fn)
```

(suite sur la page suivante)

```
# updates control version from filename, increment package version.
control.version = '%s-0'%(re.findall('Firefox Setup (.*)esr\.exe',filename)[0])
control.save_control_to_wapt()
```

Vous pouvez lancer le `update_package` dans **PyScripter** :

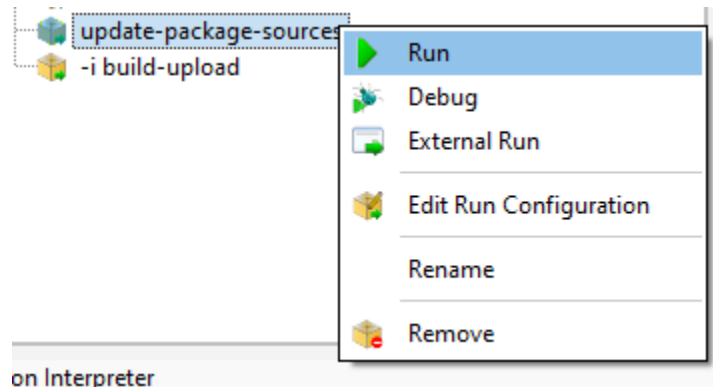


FIG. 4 – Pyscripter - Exécution d'un update-package-source

Vous trouverez de nombreux exemples d'`update_package` qui vous inspireront dans les paquets du store de Tranquil IT.

## 57.7 Exemple : déployer un logiciel portable avec WAPT

Un bon exemple de paquet applicatif WAPT est celui d'un logiciel dit *portable*. Pour cela, il faudra :

- Créer le répertoire d'installation dans C:\Program Files (x86).
- Copier l'application dans le dossier.
- Créer un raccourci sur le bureau de l'utilisateur.
- Gérer la désinstallation de l'application portable.
- Fermer l'application si elle est en cours d'exécution.

### 57.7.1 Exemple avec ADWCleaner

Tout d'abord, télécharger Adwcleaner : <https://downloads.malwarebytes.com/file/adwcleaner>.

Vous pouvez générer votre modèle de paquet, veuillez vous référer à la *documentation pour la création de paquets à partir de la console WAPT*.

Le fichier C:\waptdev\tis-adwcleaner-wapt est créé.

Vous trouverez ici un exemple de paquet portable qui prend presque toutes les fonctions WAPT d'un `setup.py` :

```
from setuphelpers import *

uninstallkey = []

exe_name = 'AdwCleaner.exe'
path_adw = makepath(programfiles, 'AdwCleaner')
```

(suite sur la page suivante)

(suite de la page précédente)

```

path_exe = makepath(path_adw, exe_name)
nameshortcut = 'AdwCleaner'

def install():
    mkdirs(path_adw)
    filecopyto(exe_name, path_exe)
    create_desktop_shortcut(nameshortcut, path_exe)

def uninstall():
    remove_tree(path_adw)
    remove_desktop_shortcut(nameshortcut, path_exe)

def audit():
    if not isfile(path_exe):
        print('File not found')
        return "OK"
    else:
        print('File Found')
        return "ERROR"

def update_package():
    wget('https://downloads.malwarebytes.com/file/AdwCleaner', exe_name)
    control.version = get_file_properties(exe_name)['FileVersion'] + '-0'
    control.save_control_to_wapt()

```

## 57.8 Créer des paquets WAPT de mises à jour Windows avec des .msu

**Indication :** Pré-requis : pour construire des paquets WAPT, *l'environnement de développement WAPT doit être installé* ;

Entre les sorties de *Patch Tuesday*, Microsoft peut publier des KB supplémentaires ou des mises à jour critiques qui devront être rapidement poussées sur les machines.

À cette fin, WAPT fournit un modèle de paquet pour les fichiers *.msu*.

Dans cet exemple, nous utilisons la KB4522355 téléchargée du site officiel Microsoft.

- télécharger le paquet MSU KB4522355 depuis le Catalogue du site de Microsoft .
- Créer un modèle de paquet WAPT à partir du fichier *.msu* téléchargé. Dans la console WAPT, cliquez sur *Outils* → *Générer un modèle de paquet*.
- Sélectionner le paquet téléchargé *.msu* et remplir les champs obligatoires.
- cliquer sur *Créer et éditer* (recommandé) pour lancer la personnalisation du paquet ;
- L'IDE du paquet WAPT est lancé en utilisant le code source du modèle prédéfini *.msu*.
- Comme d'habitude avec les paquets WAPT, tester, puis construire, puis signer, puis télécharger et enfin affecter les paquets WAPT souhaités à vos hôtes sélectionnés et c'est fait !
- Si le KB est groupé avec le *Patch Tuesday* suivant, vous pourrez sélectionner les hôtes sur lesquels le paquet a été appliqué et oublier le paquet KB sur les hôtes.

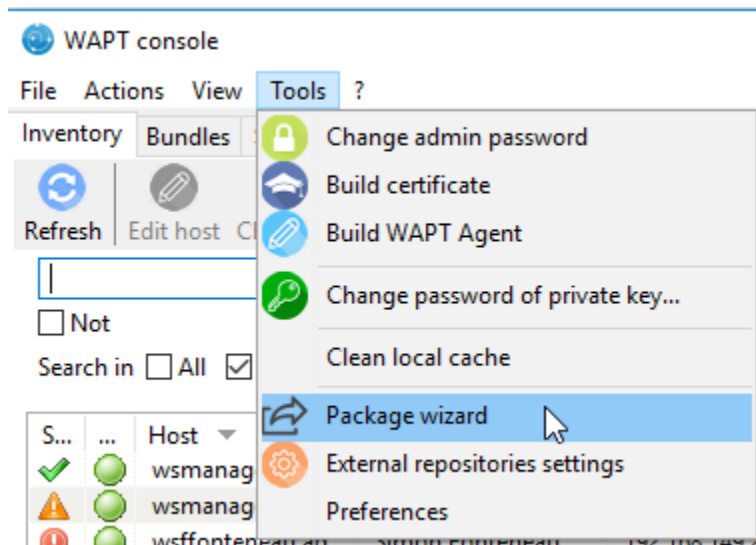


FIG. 5 – PyScripter - Menu pour la création de modèle de paquets depuis la console

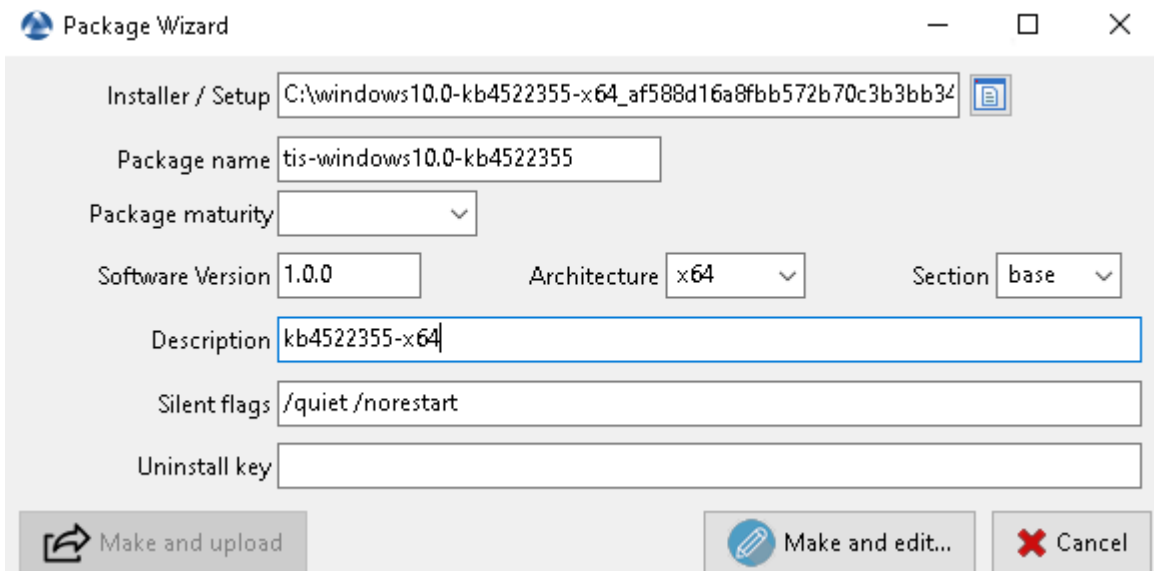


FIG. 6 – Informations requises pour la création du paquet MSU



## 57.9 Packager des paquets linux simples

Avant de commencer, nous supposons plusieurs conditions :

- Vous disposez d'une interface graphique sur votre système Linux que vous utilisez pour développer et tester des paquets.
- Vous avez installé le paquet **vscode** à partir du dépôt de Tranquil IT.
- Votre utilisateur s'appelle *linuxuser* et est membre du groupe *sudoers*.

### 57.9.1 Créer un modèle de paquet base depuis votre poste linux

- Démarrer un utilitaire de ligne de commande.
- En tant que *linuxuser*, créer un modèle de paquet WAPT.

```
wapt-get make-template <template_name>
```

**Avertissement : Ne pas lancer cette commande en tant que root ou avec sudo.**

Lorsque vous créez un modèle, il y aura plusieurs fichiers dans le dossier `.vscode` à l'intérieur de votre dossier de développement de paquets :

- `settings.json`;
- `launch.json`;

Exemple avec **TightVNC** :

```
wapt-get make-template "tis-vlc"

Using config file: /opt/wapt/wapt-get.ini
Template created. You can build the WAPT package by launching
/opt/wapt//wapt-get.py build-package /home/linuxuser/waptdev/tis-vlc-wapt
You can build and upload the WAPT package by launching
/opt/wapt//wapt-get.py build-upload /home/linuxuser/waptdev/tis-vlc-wapt
```

**Indication :** Tous les paquets sont stockés dans le répertoire personnel de *linuxuser* (le home de l'utilisateur actuellement connecté).

VSCoDe se charge et ouvre le projet de paquet.

- Vérifier le contenu du fichier `control`.  
Vous devez donner une **description** à votre paquet, et renseigner le **target\_os** et la **version** de votre paquet.

**Indication :** `os_target` pour unix est *linux*

**Avertissement :** Le numéro de version dans votre fichier `control` doit commencer à 0, et non le numéro de version du logiciel car nous ne savons pas précisément à partir de apt/yum repo quelle sera la version du logiciel.

- Fichier `control` d'origine.

```

File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
  setup.py 9+
TIS-VLC-WAPT
  .vscode
  WAPT
  setup.py 9+
setup.py x
1 # -*- coding: utf-8 -*-
2 from setuphelpers import *
3
4 uninstallkey = []
5
6 def install():
7     pass
8     # put here what to do when package is installed on host
9     # implicit context variables are WAPT, basedir, control, user, params, run
10
11 def uninstall():
12     pass
13     # put here what to do when package is removed from host
14     # implicit context variables are WAPT, control, user, params, run
15
16 def session_setup():
17     print('Session setup for %s' % control.asrequirement())
18     # put here what to do when package is configured inside a user session
19     # implicit context variables are WAPT, control, user, params
20
21 def update_package():
22     pass
23     # put here what to do to update package content with newer installers.
24     # launched with command wapt-get update-package-sources <path-to-wapt-directory>
25     # implicit context variables are WAPT, basedir, control, user, params, run
26     # if attributes in control are changed, they should be explicitly saved to package file with control.save_control_to_wapt()
27
28 def audit():
29     pass
30     # put here code to check periodically that state is matching expectations
31     # return "OK", "WARNING" or "ERROR" to report status in console.
32     # all print statement are reported too
33     return "OK"

```

Fig. 7 – Ouverture du VSCode avec le focus sur le fichier `setup`

```

package      : tis-vlc
version      : 0-0
architecture : all
section      : base
priority     : optional
maintainer   : user
description   : automatic package for vlc

```

— Fichier control modifié :

```

package      : tis-vlc
version      : 0
architecture : all
section      : base
priority     : optional
maintainer   : Tranquil-IT Systems
description   : VLC for linux
target_os    : linux
min_wapt_version : 1.8

```

**Note :** Il est à noter qu'une sous-version `-1` a été ajoutée. Il s'agit de la version de packaging du paquet WAPT. Il permet au développeur de paquets de publier plusieurs versions de paquets WAPT d'un même logiciel, ce qui est très utile pour un développement très rapide et itératif.

— Changer le code du fichier `setup.py` en conséquence.

```

:emphasize-lines: 8
# -*- coding: utf-8 -*-
from setuphelpers import *

uninstallkey = []

def install():
    apt_install('vlc')

```

— Enregistrer le paquet.

## 57.9.2 Gérer la désinstallation

— Modifier le fichier `setup.py` avec une procédure de désinstallation.

```

def uninstall():
    apt_remove('vlc')

```

— Lancer un `remove` de VSCode *Run Configurations*.

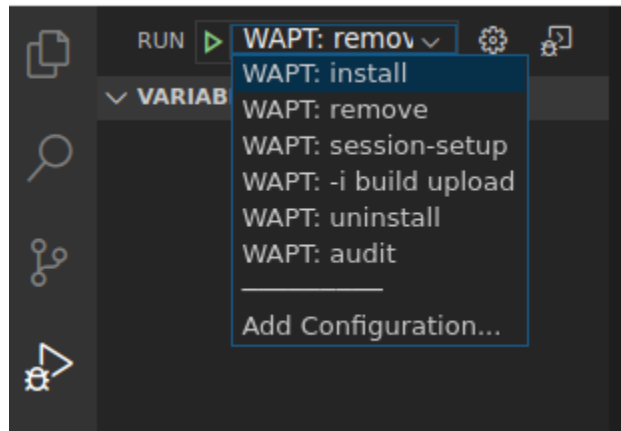


FIG. 8 – À l’issue de la désinstallation, le programme est désinstallé

— Vérifier que le logiciel a été correctement supprimé.

```

dpkg -l | grep vlc

```

**Indication :** Dans la fonction `uninstall()`, on ne peut pas appeler des fichiers contenus dans le paquet WAPT. Pour les appeler, il faudra avoir copié les fichiers dans un répertoire local de la machine lors de l’installation du paquet.

### 57.9.3 Gérer le session-setup

- Modifier le fichier `setup.py` avec un `session-setup`;  
Dans cet exemple, nous allons créer un fichier :fichier : `vlcrc` par défaut dans le profil de l'utilisateur.

```
def session_setup():
    vlrc_content="""[qt] # Qt interface
qt-notification=0
qt-privacy-ask=0
metadata-network-access=0
"""

    vlcdirc = os.path.join(os.environ['HOME'], '.config', 'vlc')
    path_vlrc = makepath(vlcdirc,'vlcrc')
    ensure_dir(vlcdirc)
    if not isfile(path_vlrc):
        with open(makepath(vlcdirc,'vlcrc')) as f:
            f.write(vlrc_content)
```

- Lancez un `session-setup` à partir de VSCode *Run Configurations*.

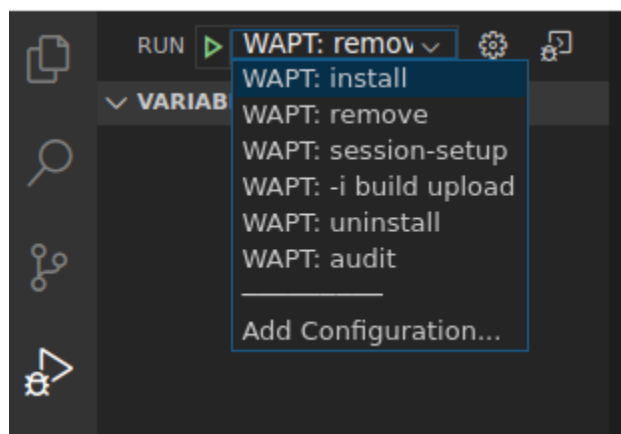


FIG. 9 – À l'issue de la désinstallation, le programme est désinstallé

### 57.9.4 Construire et téléverser le paquet

Vous trouverez votre paquet ici : `~/waptdev`.

Vous devez transférer le dossier du paquet sur la machine Windows qui possède la clé privée.

Ensuite, se référer à la *documentation pour la construire et charger le paquet depuis la console WAPT*.

## 57.10 Chiffrer des données sensibles contenues dans un paquet WAPT

**Note :** Cette partie de la documentation est déconseillée aux utilisateurs qui débutent avec WAPT.

Cette fonctionnalité est disponible uniquement dans la version **Entreprise**.

### 57.10.1 Quel est l'intérêt de faire cela ?

Dans le fonctionnement de WAPT, l'intégrité du paquet est assurée. Un paquet dont le contenu a été modifié sans avoir été re-signé sera systématiquement refusé par le client WAPT.

En revanche le contenu d'un paquet WAPT n'est pas chiffré et sera lisible de tous. Ce modèle technique de transparence apporte cependant de nombreux bénéfices.

Cela peut être gênant dans le cas d'un paquet qui contiendrait un mot de passe, une clé de licence, ou une donnée sensible.

Heureusement **nous avons une solution. !!**

### 57.10.2 Principe de fonctionnement

Lorsque qu'un agent WAPT s'enregistre auprès du serveur WAPT, il génère un couple clé privée / certificat public dans C:\Program Files (x86)\waptprivate.

- Le certificat est envoyé au serveur avec l'inventaire lors de l'enregistrement initial du client WAPT.

- La clé privée est conservée par l'agent et n'est accessible en lecture que par les *Administrateurs Locaux*.

Nous allons donc chiffrer la donnée sensible contenue dans le paquet avec le certificat appartenant à la machine.

Lors de l'installation l'agent WAPT pourra ainsi déchiffrer la donnée sensible grâce à sa clé privée.

Avec ce mode de fonctionnement le serveur WAPT et les dépôts secondaires n'ont pas connaissance de la donnée sensible.

### 57.10.3 Cas pratique

Vous trouverez ici un exemple de paquet WAPT où nous chiffrons un texte dans une fonction `update_package` puis nous déchiffrons ce texte dans la partie `install`.

Dans cet exemple, la fonction `update_package` nous permet de parcourir la base de donnée du serveur WAPT pour récupérer le certificat de chaque machine pour ensuite chiffrer le texte sensible avec celui-ci.

Le texte chiffré pour chaque machine est ensuite stocké dans un fichier `encrypt-txt.json` à la racine du paquet.

Lors de l'installation du paquet, l'agent WAPT prendra le texte chiffré et le déchiffrera avec sa clé privée.

```
# -*- coding: utf-8 -*-
from setuphelpers import *
import json
from waptcrypto import SSLCertificate
import waptguihelper
import base64

def install():
```

(suite sur la page suivante)

```
encryptlist = json.loads(open('encrypt-txt.json','r').read())
if WAPT.host_uuid in encryptlist:
    host_key = WAPT.get_host_key()
    v = base64.b64decode(encryptlist[WAPT.host_uuid])
    encrypttxt = host_key.decrypt(v).decode('utf-8')
    print(r"Here is the deciphered text: %s" % encrypttxt)
else:
    error("%s not found in encrypt-txt.json" % WAPT.host_uuid)

def update_package():
    urlserver = inifile_readstring(makepath(install_location('WAPT_is1'),'wapt-get.ini'),'global',
    ↪ 'wapt_server').replace('https://','')
    encrypttxt = input("Enter the text to be encrypted: ")
    encryptlist = {}
    credentials_url = waptguihelper.login_password_dialog('Credentials for wapt server',urlserver,
    ↪ 'admin','')
    data = json.loads(wgets('https://%s:%s@%s/api/v1/hosts?columns=host_certificate&limit=10000' % (
    ↪ credentials_url['user'],credentials_url['password'],urlserver)))
    for value in data['result']:
        if value['host_certificate']:
            host_cert = SSLCertificate(cert_string = value['host_certificate'])
            encryptlist[value['uuid']] = base64.b64encode(host_cert.encrypt(encrypttxt.encode('utf-8'
    ↪ '))).decode('utf-8')
            print(value['computer_fqdn'] + ':' + value['uuid'] + ':' + encryptlist[value['uuid']])
    open('encrypt-txt.json','w').write(json.dumps(encryptlist))
```

**Attention :** La sortie python (log install du paquet) est accessible en lecture aux utilisateurs de la machine, **vous ne devez donc pas afficher le texte déchiffré avec un print lors de l'installation.**

---

## Utiliser des IDE différents pour développer les paquet WAPT

---

Si vous êtes habitué(e) à travailler avec un autre *IDE*, vous pouvez être soulagé maintenant car WAPT supporte d'autres éditeurs de développement intégrés.

Certains éditeurs de code sont pris en charge en natif :

- PyScripter ;
- VSCode ;
- VSCodium ;

D'autres éditeurs peuvent être sélectionnés et seront lancés lorsque vous créez un nouveau modèle pour un paquet WAPT à partir de la console WAPT.

---

**Note :** Utiliser un IDE supporté lancera le projet de paquet WAPT avec une configuration de débogage valide.

---

### 58.1 Sur Windows

Pour configurer un autre éditeur pour WAPT, vous devez modifier l'attribut `editor_for_packages` dans la section `[global]` du fichier de configuration `%LOCALAPPDATA%\waptconsole\waptconsole.ini` de votre console WAPT.

TABLEAU 1 – Méthodes alternatives pour l'édition de paquets WAPT sur Windows

Nom de l'éditeur de code	editor_for_packages value
PyScripter	None
Microsoft Visual Studio Code	<b>vscode</b> ou <b>code</b>
Microsoft Visual Studio Codium	<b>vscodium</b> ou <b>codium</b>

Exemple de configuration de `waptconsole.ini` :

```
[global]
...
editor_for_packages=vscode
```

## 58.2 Sur Linux / macOS

Pour configurer un autre éditeur pour WAPT, vous devez modifier l'attribut `editor_for_packages` dans la section `[global]` du fichier de configuration `/opt/wapt/wapt-get.ini` de votre agent WAPT.

Par défaut, si l'attribut `editor_for_packages` est vide, le WAPT essaiera de lancer (dans cet ordre) :

- **vsodium**;
- **vscode**;
- **nano**;
- **vim**;
- **vi**.

TABLEAU 2 – Méthodes alternatives pour l'édition de paquets WAPT sous Linux

Nom de l'éditeur de code	editor_for_packages value
Microsoft Visual Studio Code	<b>vscode</b> ou <b>code</b>
Microsoft Visual Studio Codium	<b>vsodium</b> ou <b>codium</b>
Nano	<b>nano</b>
Vim	<b>vim</b>
Vi	<b>vi</b>

```
[global]
...
editor_for_packages=vim
```



---

## Configurer WAPT pour utiliser un éditeur de code personnalisé

---

### 59.1 Sur Windows

Des éditeurs de code personnalisés peuvent être utilisés, par exemple **Notepad++** ou **PyCharm**.

TABLEAU 1 – Utiliser un éditeur de texte personnalisé sous Windows

Nom de l'éditeur de code	editor_for_packages value
Notepad++	C:\Program Files\Notepad++\notepad++.exe <i>setup_filename</i>
PyCharm	C:\Program Files\JetBrains\PyCharm Community Edition 2019.3.2\bin\pycharm64.exe <i>wapt_sources_dir</i>

[global]

...

```
editor_for_packages=C:\Program Files\Notepad++\notepad++.exe {setup_filename}
```

### 59.2 Sur Linux / macOS

Des éditeurs de code personnalisés peuvent être utilisés, par exemple **PyCharm**.

TABLEAU 2 – Utiliser un éditeur de texte personnalisé sur linux / macos

Nom de l'éditeur de code	editor_for_packages value
PyCharm	/opt/pycharm/bin/pycharm_x64 <i>wapt_sources_dir</i>

```
[global]
...
editor_for_packages=/opt/pycharm/bin/pycharm_x64 {wapt_sources_dir}
```

## 59.2.1 Arguments personnalisés

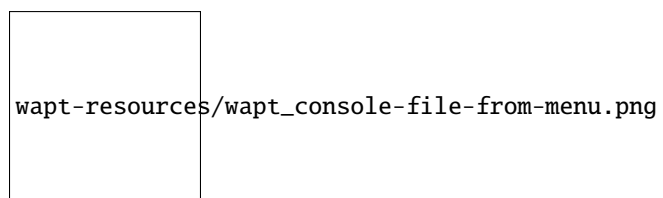
TABLEAU 3 – Les arguments peuvent être passés dans la commande `editor_for_packages`

Argument	Description
<code>{setup_filename}</code>	Lance l'éditeur de code personnalisé et édite le fichier WAPT <code>setup.py</code>
<code>{control_filename}</code>	Lance l'éditeur de code personnalisé et modifie le fichier <code>control</code> des paquets WAPT
<code>{wapt_sources_dir}</code>	Lance l'éditeur de code personnalisé et ouvre le dossier du paquet WAPT
<code>{wapt_base_dir}</code>	Lance l'éditeur de code personnalisé et ouvre le dossier d'installation WAPT

Cette page détaille les diverses fenêtres de la console WAPT.

Détail des onglets de la barre de menu

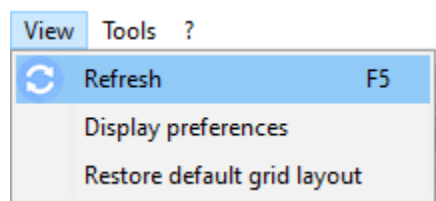
### 60.1 Onglet Fichier (File)



L'action ferme la console WAPT. Le raccourci clavier est Ctrl+Q.

### 60.2 Onglet Affichage (View)

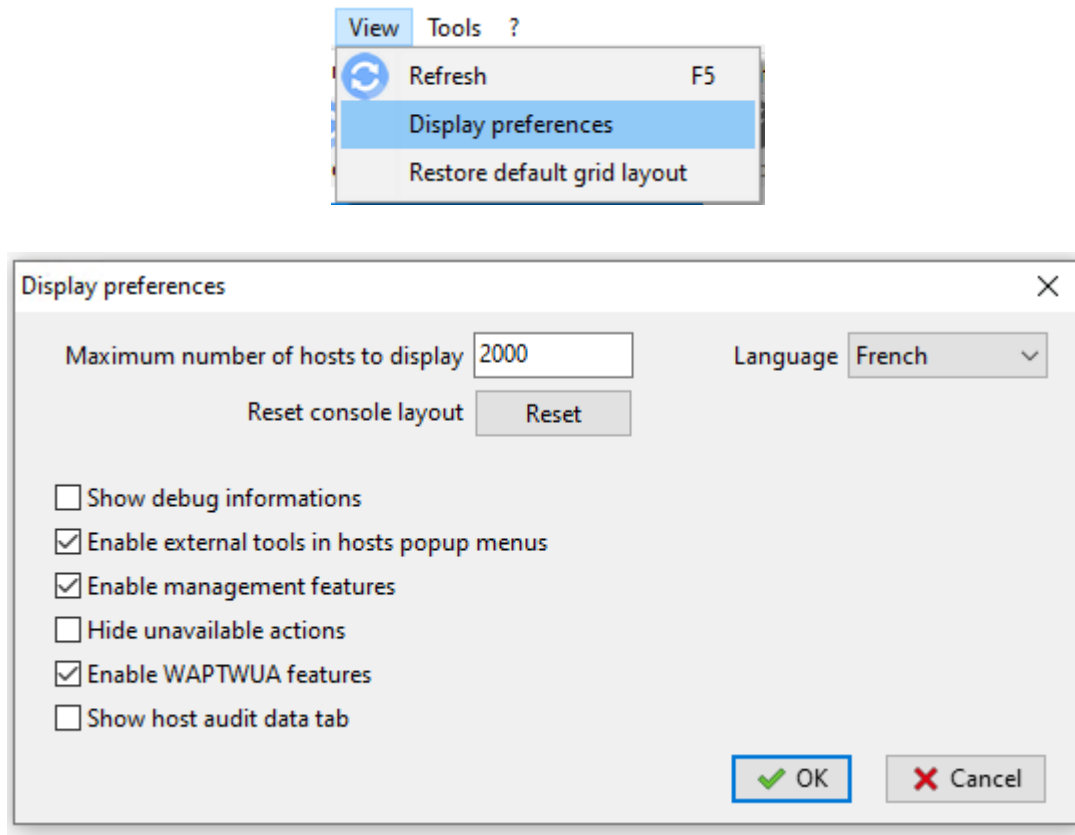
#### 60.2.1 Rafraîchir



Rafraîchit l'affichage de la console WAPT.

La raccourci clavier est F5.

### 60.2.2 Afficher les préférences

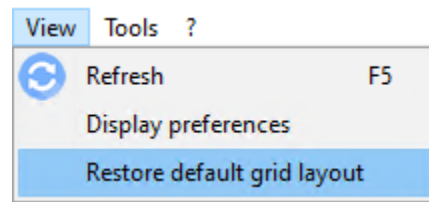


Cette fenêtre vous permet de configurer vos propres préférences pour la console WAPT.

TABLEAU 1 – Liste des actions disponibles dans préférences

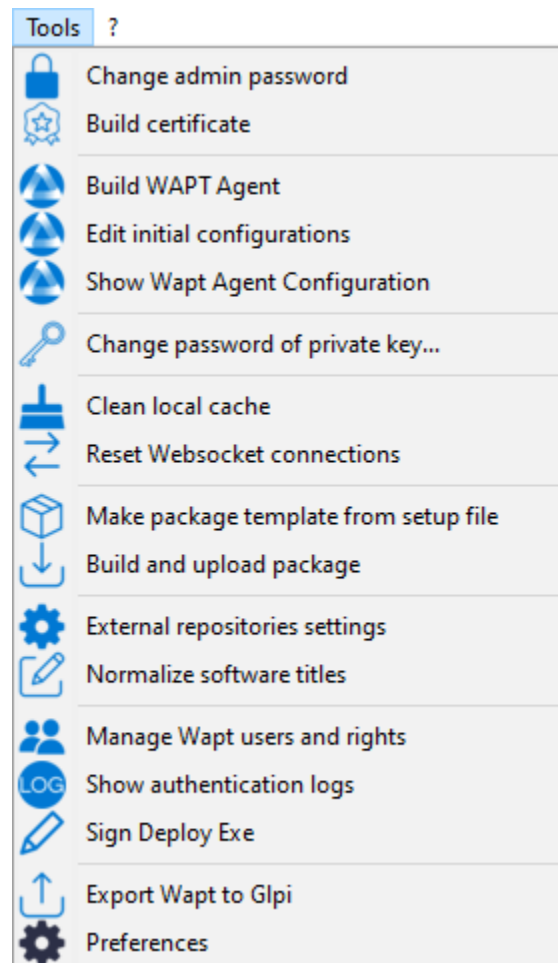
Nom	Description	Exemple
Nombre maximum d’hôtes à afficher	Nombre maximum d’hôte dans l’Inventaire	2000
Langue	Montre la langue locale de la console WAPT	Français/Anglais/Allemand
Réinitialiser la mise en page de la console	Charger les paramètres par défaut	
Montrer les informations de debug	Ajouter les informations de debug en bas de la console WAPT	False
Active les outils externe dans les options du menu hôte	Affiche les outils Windows avec un clic-droit	True
Cacher les actions indisponibles	Cacher si l’action n’est pas possible pour l’utilisateur courant	False
Active les fonctionnalités WAPTWUA	Affiche l’onglet Windows Update	True
Affiche l’onglet des données d’audit de l’hôte	Affiche l’onglet d’inventaire de l’hôte	False

### 60.2.3 Restaurer l'affichage de la grille par défaut

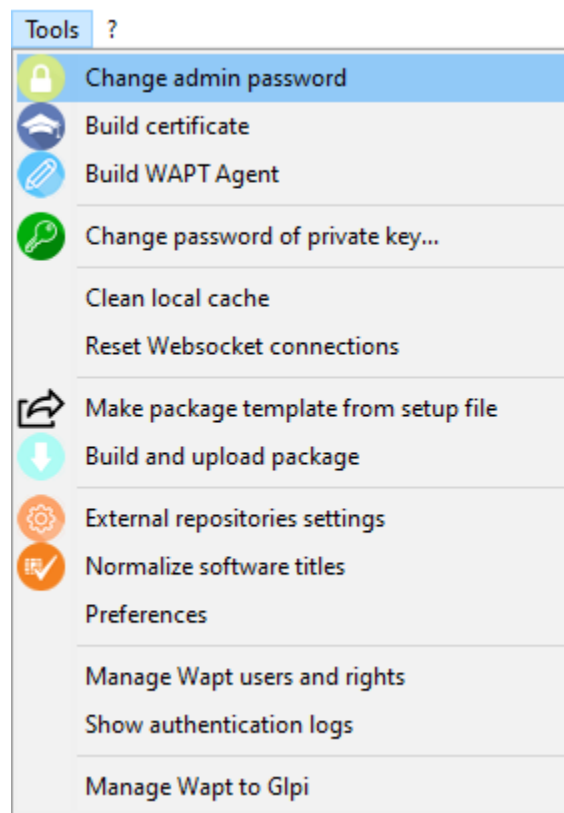


Ré-applique l'affichage par défaut. Par exemple, vous pouvez supprimer la colonne *Hardware inventory ajouté dans cette section de la documentation*.

## 60.3 Onglet outils



### 60.3.1 Changer le mot de passe Superadmin du serveur WAPT



Permet de mettre à jour le mot de passe Super Admin créé pendant le postconf.

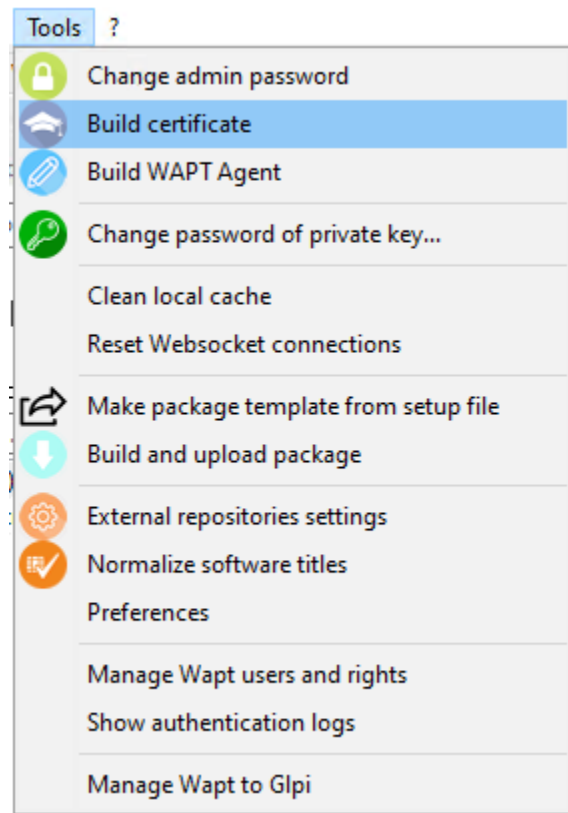
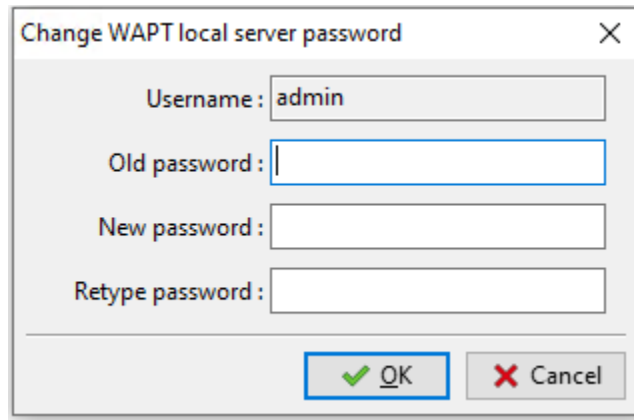
Change le mot de passe du Serveur WAPT, donnez l'ancien mot de passe puis ajoutez le nouveau.

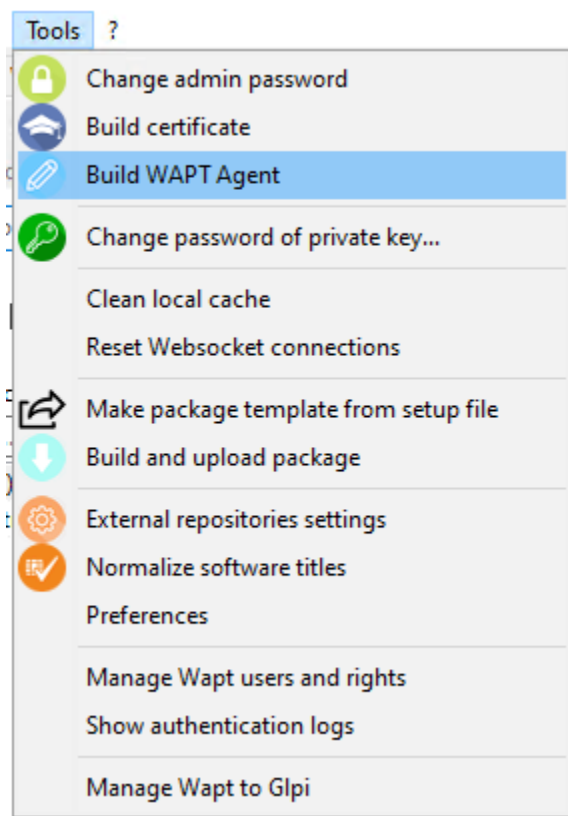
### 60.3.2 Construire un certificat Administrateur

Référez-vous à la documentation *générer le certificat Administrateur pour signer des paquets WAPT*.

### 60.3.3 Construire l'agent WAPT

Référez-vous à la documentation *construire l'installateur de l'agent WAPT*.





### 60.3.4 Changer le mot de passe de la clé privé de l'Administrateur

Permet de mettre à jour le mot de passe de la *clé privée*.

Pour changer le mot de passe de la clé privée, indiquez l'ancien mot de passe et renseignez le nouveau.

### 60.3.5 Nettoyer le cache local

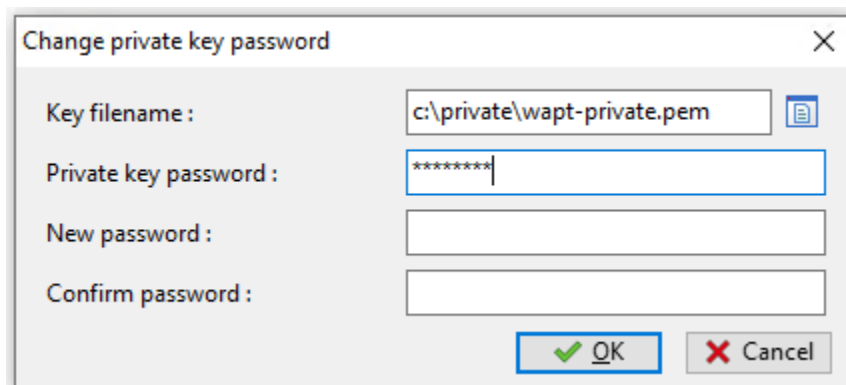
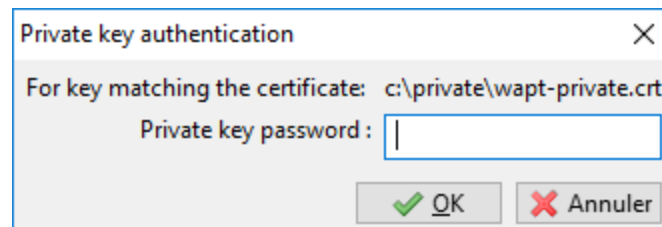
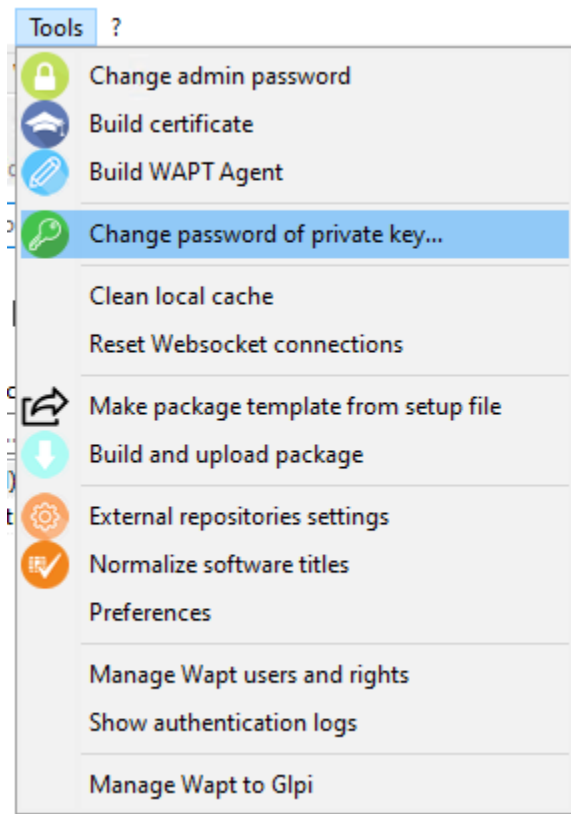
Lors de l'import d'un paquet depuis Internet, la console WAPT télécharge le paquet dans `%appdata%\local\waptconsole\cache`.

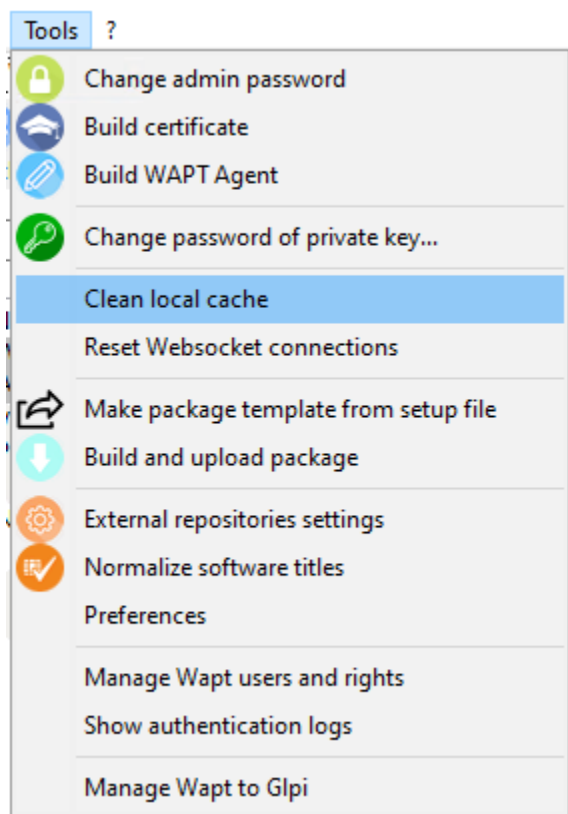
Pour nettoyer le cache local et libérer de l'espace disque, cliquez sur *Outils* → *Nettoyer le cache local*.

### 60.3.6 Réinitialiser les connexions Websocket

Utilisez cette méthode si vous avez redémarré le service waptserver sans avoir redémarré le service web **Nginx**.







### 60.3.7 Créer un modèle de paquet WAPT depuis un fichier setup

Pour plus d'information, référez-vous à la documentation pour *créer un modèle de paquet depuis la console WAPT*.

### 60.3.8 Construire et téléverser un paquet WAPT depuis la console vers un dépôt

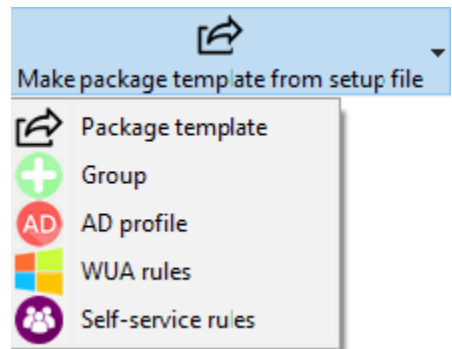
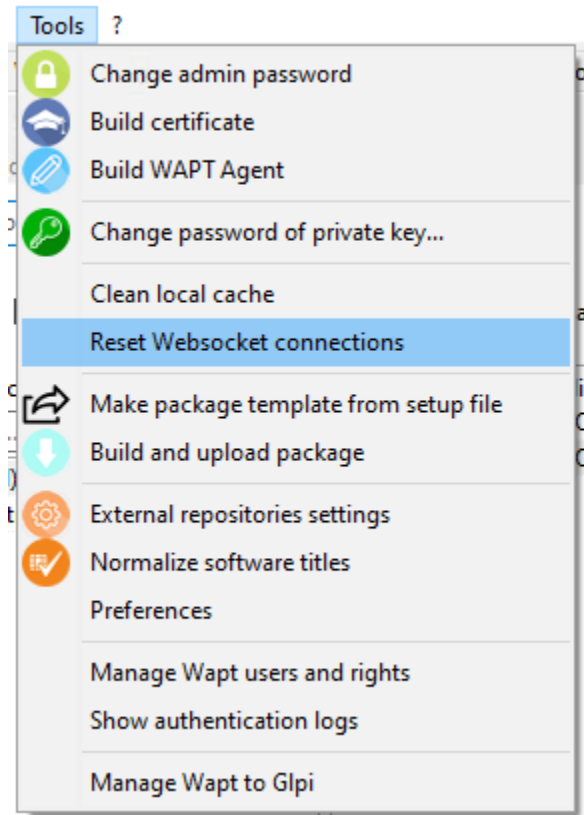
Pour plus d'informations, référez-vous à la documentation *construire un paquet et l'envoyer au serveur WAPT*.

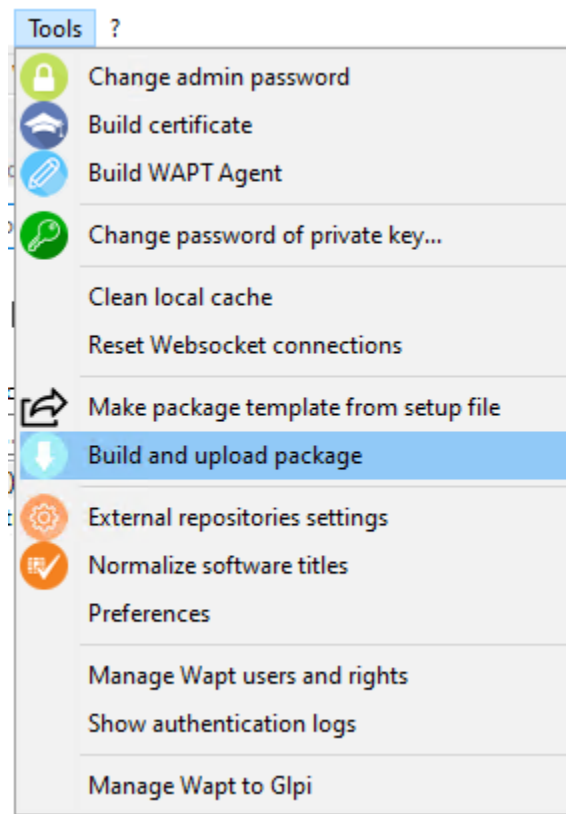
### 60.3.9 Paramètres des dépôts externes

Il est possible d'ajouter d'autres dépôts externes pour la récupération de paquet venant de sources différentes.

TABLEAU 2 – Liste des actions disponibles applicables aux dépôts

Nom	Description	Exemple
Nom du dépôt	Nom du dépôt dans la liste	wapt-template
Inscrire un nouveau dépôt	Effacer les certificats	
Désinscrire le dépôt	Supprimer le dépôt sélectionné	
Dépôt de paquets externe	adresse du dépôt	<a href="https://store.wapt.fr/wapt">https://store.wapt.fr/wapt</a>
Rechercher les certificats	Télécharger les certificats du dépôt	
proxy http à utiliser (si besoin)	adresse du proxy	<a href="http://proxy.mydomain.lan">http://proxy.mydomain.lan</a>
Paramètres avancés	Affichage des paramètres avancés	Coché
Vérifier le certificat HTTPS du serveur	Utilisation des certificats HTTPS	Coché





### 60.3.10 Normaliser les noms de logiciels

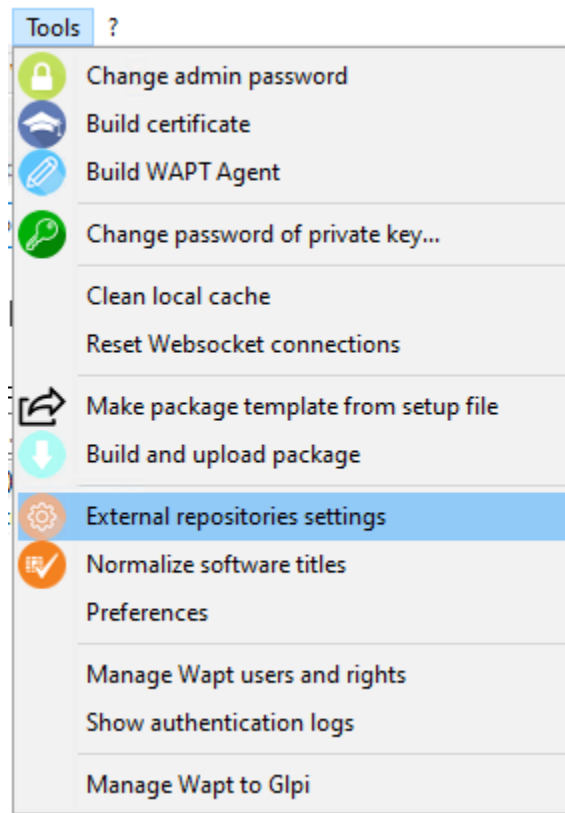
Pour plus d'informations, référez-vous à la documentation *normaliser le nom des logiciels*.

### 60.3.11 Configurer les préférences de la console WAPT

Pour changer les paramètres de la console WAPT, aller dans *Tools* → *Preferences*.

#### Configuration basique

L'onglet basique pour les options de configuration basiques ;



Repositories settings

Repository name:

External packages repository:

http proxy to use (if needed):

Advanced parameters  Check HTTPS server certificate

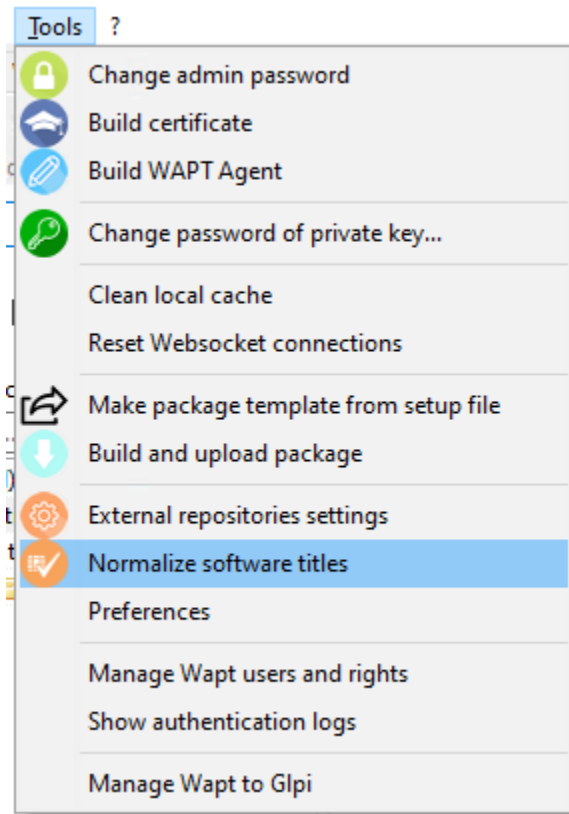
Path to https CA certificates:

Check package signature

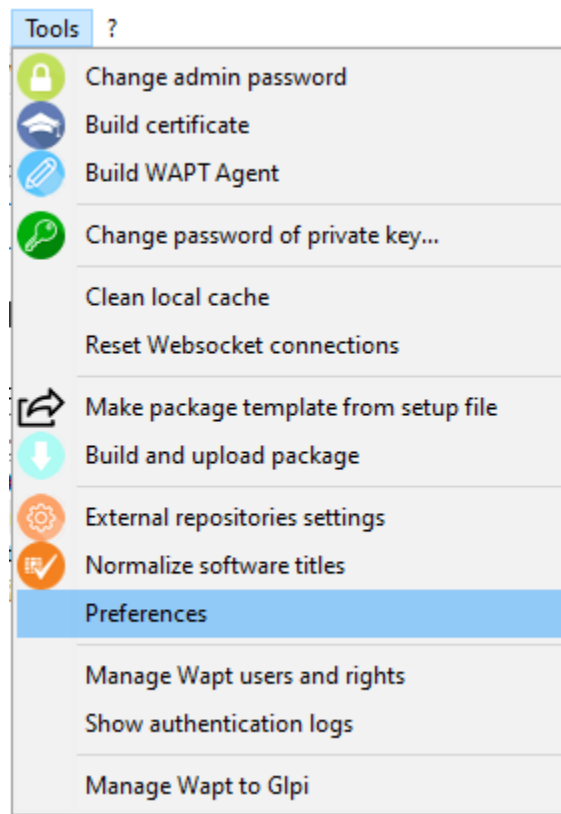
Authorized external keys dir:

Client auth certificate path:

Client auth key path:



Arguments	Description	Exemple
Adresse du serveur WAPT ou son nom	URL (IP ou FQDN) du serveur WAPT	<i>srvwapt.mydomain.lan.</i>
<i>Check and set</i>	Check whether the server exists and set this configuration. Write the WAPT Server URL to <i>waptconsole.ini</i>	
Manual override	Used for bypass automatic generation.	Coché
URL to the main WAPT repository	Set repository address of WAPT (only if <i>Manual override</i> is checked).	<a href="http://srvwapt.mydomain.lan/wapt/">http://srvwapt.mydomain.lan/wapt/</a>
WAPT server URL	Set the WAPT Server address (only if <i>Specify manually</i> is checked).	<i>https://srvwapt.mydomain.lan/</i>
Verify the HTTPS server certificate	Indicates whether the HTTPS certificate must be verified.	Coché
Path to CA certificates bundle	Path to the CA bundle of certificates that will allow certificates to be verified. Visit <i>the documentation on activating HTTPS verification</i> .	C:\Program Files (x86)\wapt\lib\site-packages\certifi\cacert.pem
<i>Get Server https Certificate</i>	Retrieve the https certificate chain from the WAPT Server.	
WAPT packages prefix	Prefix that is given to packages during replication.	tis
Path to personal certificate	Path to the certificate associated with the private key used to sign packages.	C:\private\mykey.crt
Licence directory	Path to the licence for Enterprise version. Set (empty) by default in WAPT install folder. It is possible to set another path.	X:\licence
<i>Check matching private key</i>	Check whether the <i>.crt</i> certificate matches the <i>.pem</i> private key.	
<i>New private key and certificate</i>	Create a new <i>private key / certificate</i> pair.	
Afficher le fichier de configuration	Open <code>%appdata%\Local\waptconsole\waptconsole.ini</code> file with a text editor.	Tous les paramètres de la console WAPT



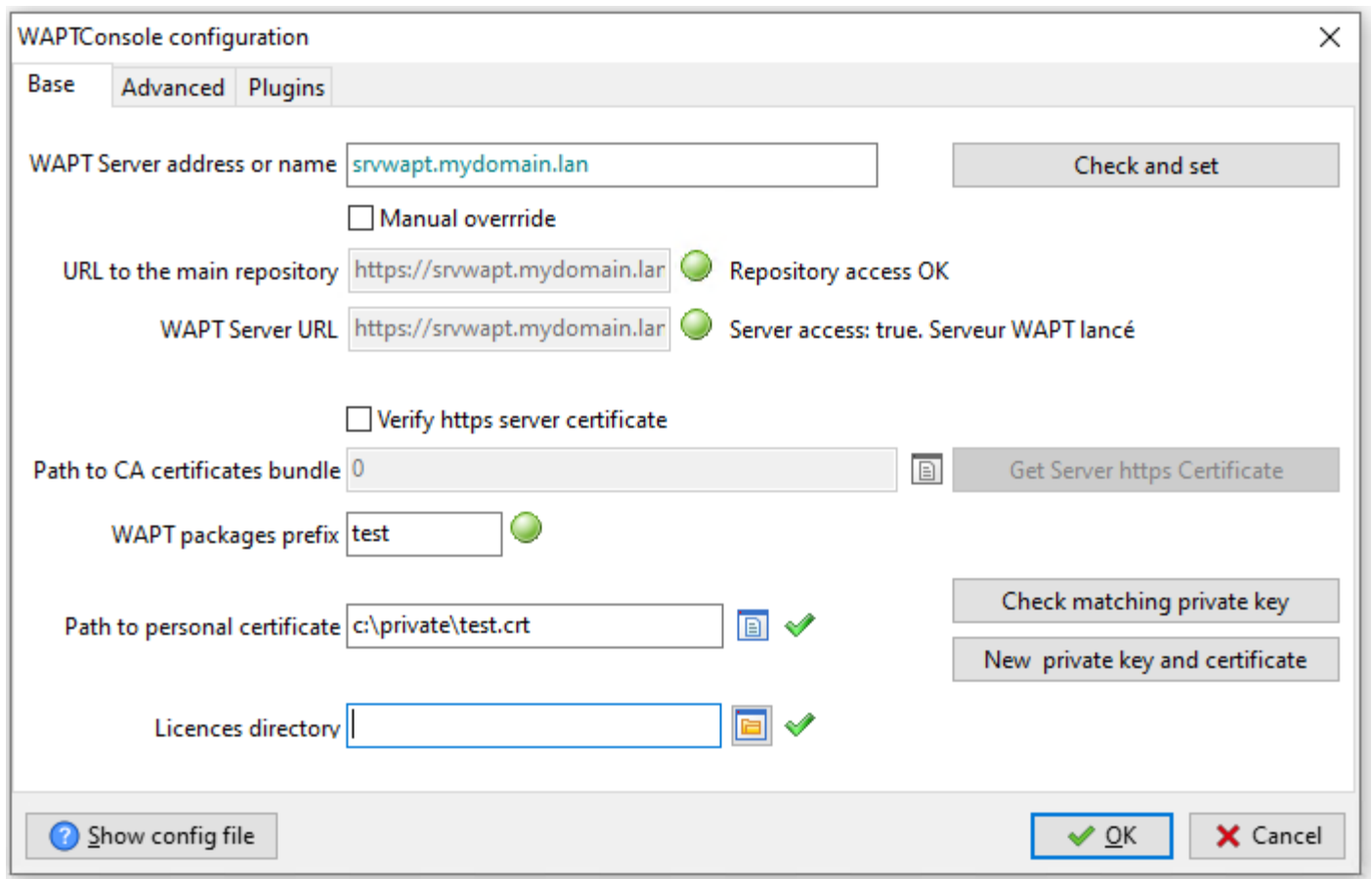
---

**Indication :** Le bouton *Récupération certificat serveur https* télécharge le certificat HTTPS du serveur dans `WAPT\ssl\server` et dit à la console WAPT de vérifier les connexions HTTPS avec le bundle de certificats. Cette méthode est appelé du *L'épinglage de certificat*. Avant de télécharger le certificat HTTPS, vous devez être sûr que vous vous connectez au bon serveur.

---

## Configuration avancé


Onglet Avancé pour les options de configuration avancé ;





WAPTConsole configuration


Base Advanced Plugins


Base directory for package development  


http proxy to use (if needed)

Activates proxy when accessing the main repository


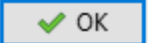
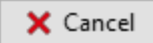
Activate proxy when accessing the WAPT server

Default package maturity  

Client auth certificate path  

Client auth key path  

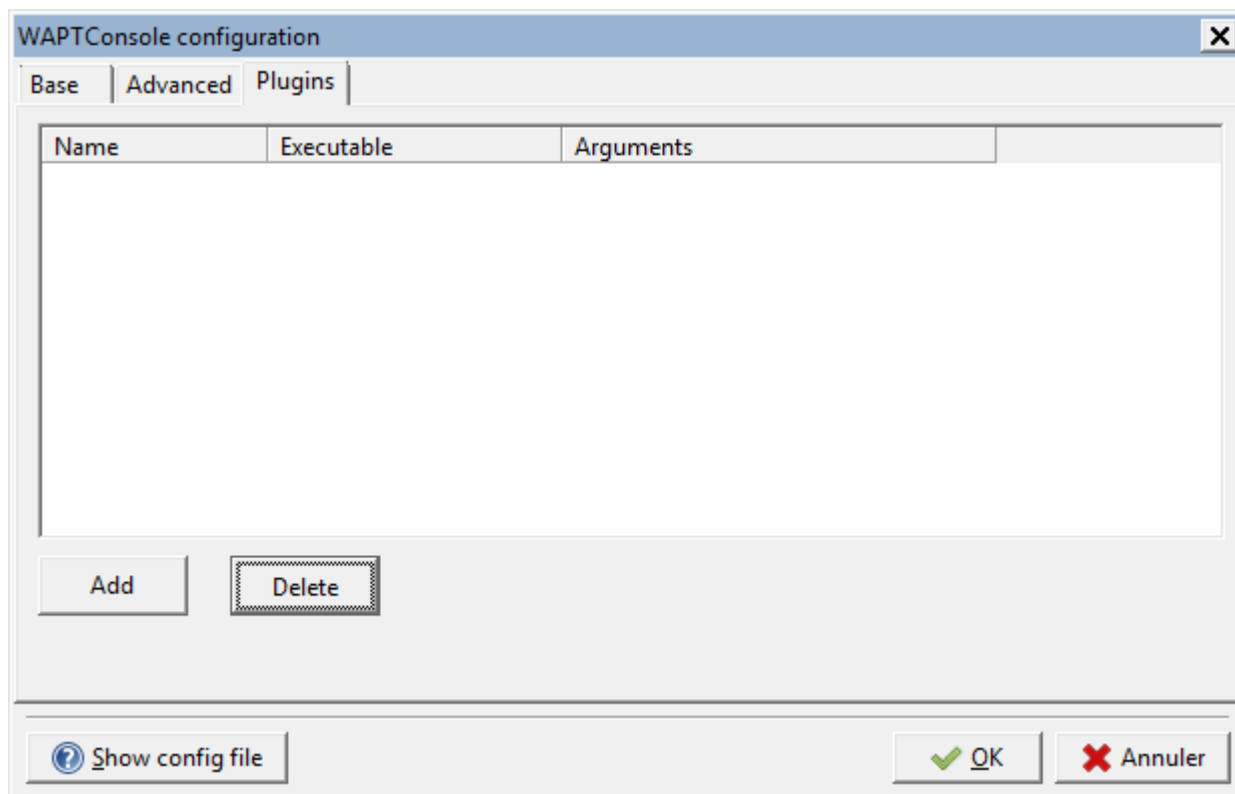
Editor for packages

 Show config file  

Arguments	Description	Exemple
Répertoire de base pour le développement de paquet	Indique le chemin vers le répertoire stockant les paquets développés.	C:\waptddev
proxy http à utiliser (si besoin)	Indique un serveur proxy qui sera utilisé par la console pour accéder au dépôt WAPT ou au serveur WAPT.	http://srvproxy.mydomain.lan:8080
Active le proxy pour l'accès au dépôt principal	Active les paramètres proxy pour l'accès au dépôt WAPT.	Coché
Active le proxy pour accéder au serveur WAPT	Active le proxy pour accéder au serveur WAPT.	Coché
Maturité des paquets par défaut	Maturité par défaut pour les paquets importés.	PROD
Chemin certificat SSL client	Chemin du certificat utilisant <i>L'authentification SSL Client</i> .	Vide
Chemin clé SSL client	Chemin de la clé utilisant <i>L'authentification SSL Client</i> .	Vide
Editeur pour les paquets	Editeur de développement par défaut pour importer les paquets WAPT.	PyScripter
Afficher le fichier de configuration	Ouvre le fichier waptconsole.ini dans %appdata%\Local\waptconsole	Tous les paramètres de la console WAPT

### Outils externes

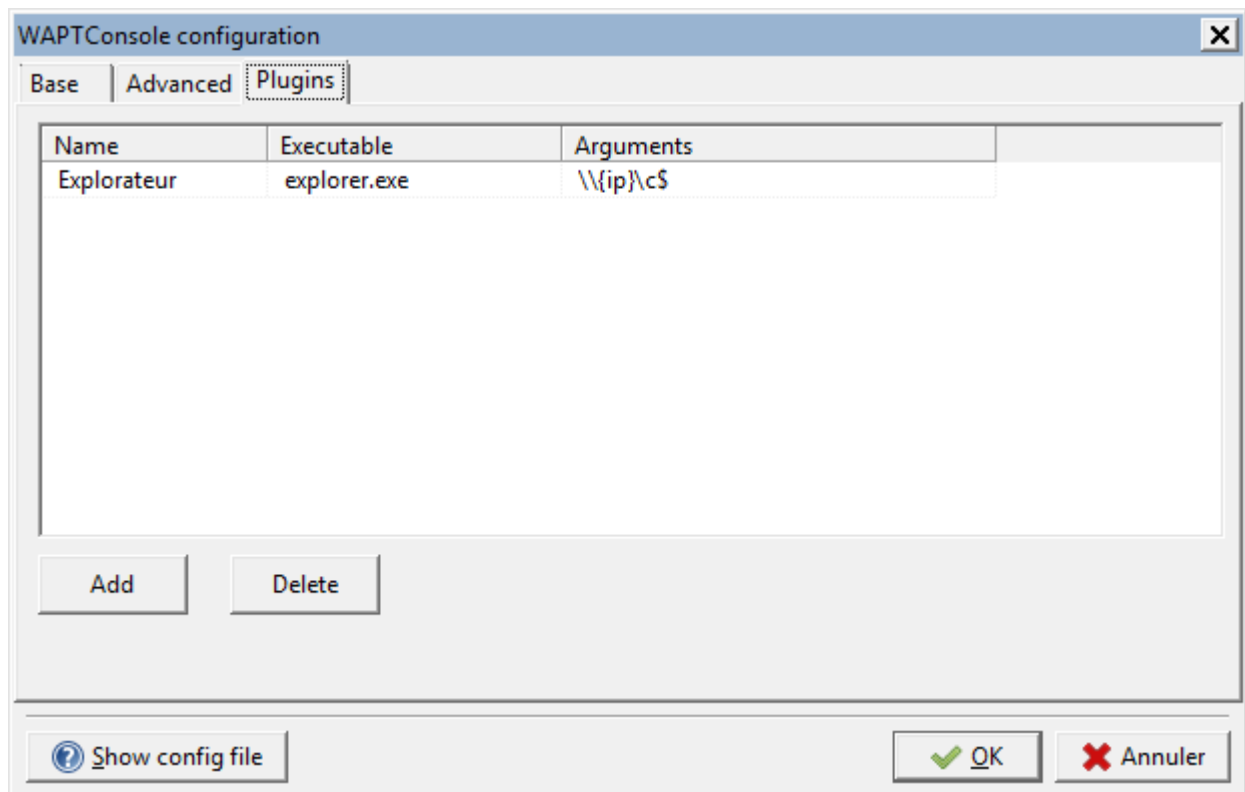
Les outils externes vous permettent d'ajouter des actions personnalisées qui ne sont pas nativement disponibles avec WAPT.



Cliquez sur *Ajouter* pour ajouter un plugin, puis éditer la colonne correspondante.

Colonne	Description
Nom	Le nom qui apparaîtra dans le menu.
Exécutable	Chemin de l'exécutable qui se lancera après le clic.
Arguments	Les arguments passés à l'exécutable. Certaines variables peuvent être utilisées comme {{ ip }}, {{ uuid }} ou {{ computer_fqdn }}.
Afficher le fichier de configuration	Ouvrez le fichier waptconsole.ini dans %appdata%/Local/waptconsole.

Les plugins vont alors apparaître dans le menu :



## 60.4 L'onglet ?

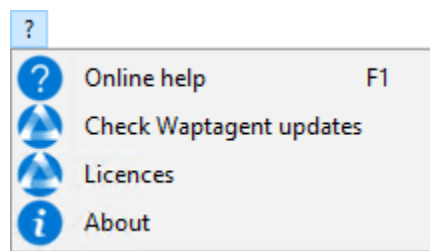


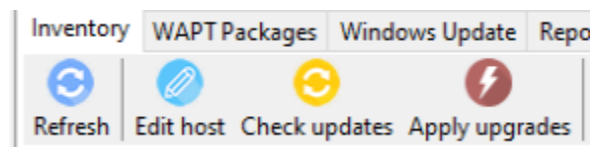
FIG. 1 – Onglet plus d'info



TABLEAU 3 – Liste des actions disponibles

Colonne	Description	Exemple
Status	Statut de validité de la licence	<i>Ok</i>
Nombre	Nombre de poste compris dans la licence	<i>Fermer</i>
No de licence	Numéro de licence	12345678-aaa-bbb-ccc-123456789acb
Concédé à	Détenteur de la licence	pdegalles
Date	Date de création de la licence	2021-01-12 00 :00
Début validité	Date de début de la licence.	2021-01-12 00 :00
Fin de validité	Date de fin de la licence.	2022-01-12 00 :00
Email de contact	Email de contact du détenteur de la licence.	<a href="mailto:pdegalles@paysdegalles.org">pdegalles@paysdegalles.org</a>
Message	Message affiché lorsque la licence a un statut <i>ERREUR</i> .	<i>SSLVerifyException : SSL signature verification failed for certificate</i>
Nom du fichier	Chemin à utiliser pour obtenir la licence avec son nom	C:\Program Files (x86)\wapt\licences\ pdegalles-12345678-aaa-bbb-ccc-123456789acb.lic



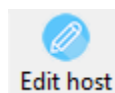


## 61.1 Rafraîchir

*Rafraîchir* rafraîchit l’affichage de la console WAPT.

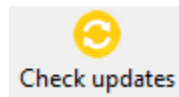
La raccourci clavier est F5.

## 61.2 Modifier la machine



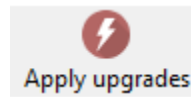
*Edit host* permet d’éditer le paquet machine.

## 61.3 Vérification des mises à jours



*Check updates* permet de vérifier les mises à jour de la ou des machine(s) sélectionnée(s).

## 61.4 Lancer les installations



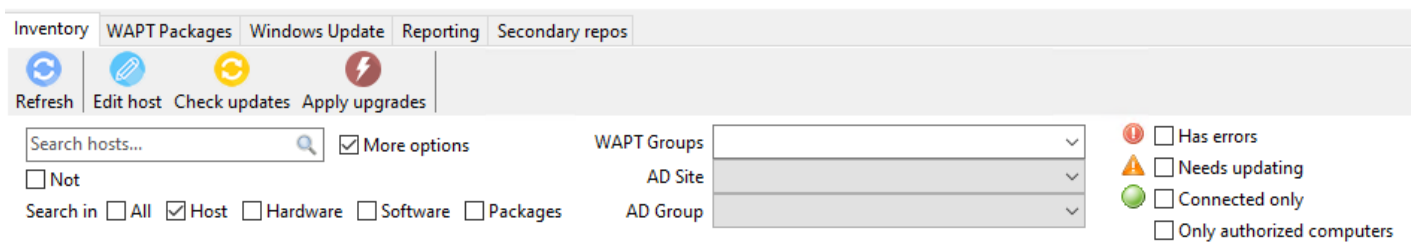
*Apply updates* permet d'appliquer les mises à jour de la ou des machine(s) sélectionné(s).



---

## Effectuer une recherche globale sur tous les postes

---



Les filtres permettent de faire une recherche suivant *différents critères*.



La console WAPT n'est pas directement connecté à l'Active Directory mais reconstruit en temps réel la structure depuis les informations des clients.

Si la structure n'est pas à jour il faut exécuter une *Vérification des mises à jour* sur un *client* ou sur l'*OU*.

### 63.1 Inclure les postes des sous-dossiers

Include computers from subfolders

Cette option permet d'afficher tous les postes présents dans les dossiers et sous-dossiers dans les OU.

Si non coché, seuls les postes présents dans l'OU sélectionnée seront affichés dans la console.

### 63.2 Rechercher dans les Unités Organisationnelles

Search OU...

Cette option vous permet de rechercher dans les dossier d'OU.

---

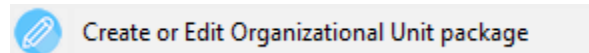
**Indication :** Les filtres fonctionnent avec les *expressions régulières*.

---

### 63.3 Afficher les Unités Organisationnelles

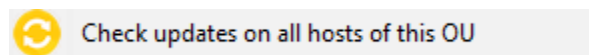
Développer les dossiers d'Unités Organisationnelles vous permet d'afficher les OU imbriquées.

### 63.4 Créer ou éditer le paquet de l'Unité Organisationnelle



Cette option permet de *modifier le paquet Unit* sélectionné.


### 63.5 Vérifier les mises à jour des machines de cette Unité Organisationnelle









Cette option permet de *vérifier les mises à jour* de toutes les machines de l'OU.

### 63.6 Appliquer les mises à jours de tous les hôtes de cette Unité Organisationnelle

Cette option permet d'*appliquer les mises à jour* de toutes les machines de l'OU.

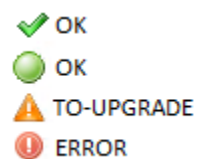
 Apply upgrades on all hosts of this OU



Status	Reachable	Audit status	WUA	Description
 OK	 OK	 OK		
 ERROR	 OK	 OK		

L'inventaire central affiche tous les postes selon les filtres sélectionnés.

### 64.1 Statut de l'hôte



Les différentes colonnes peuvent avoir différents statut.

Status	Description
<i>OK</i>	L'hôte ne remonte pas de soucis.
<i>OK</i>	L'hôte est connecté.
<i>TO-UPGRADE</i>	L'hôte à une installation / mise à jour en attente.
<i>ERROR</i>	L'hôte à des erreurs.

## 64.2 Sous onglet de l'hôte

Status (host\_status)  
Reachable (reachable)  
Audit status (audit\_status)  
WUA (waptwua\_status.status)  
Host (computer\_fqdn)  
Description (description)  
Manufacturer (manufacturer)  
IP addresses (connected\_ips)  
Platform (platform)  
Operating system (os\_name)  
Model (productname)  
Serial number (serialnr)  
Last seen on (last\_seen\_on)  
Logged in users (connected\_users)  
MAC address (mac\_addresses)  
WAPT version (wapt\_version)  
uuid (uuid)  
purchase\_order (purchase\_order)  
purchase\_date (purchase\_date)  
attributes (attributes)  
Domain controller (host\_info/domain\_controller)  
Groups (depends)  
Domaine (host\_info/domain\_name)  
Listening address (listening\_address)  
Domain controller address (host\_info/domain\_controller\_address)  
AD DN (host\_info/computer\_ad\_dn)  
Last logged on user (last\_logged\_on\_user)  
Registration Method (registration\_auth\_user)  
Repositories (repositories)

Toutes les colonnes ne sont pas affichées par défaut, il est possible d'en ajouter par un clic droit sur l'en-tête de l'inventaire.



Status	Description
<i>Status (host_status)</i>	Statut de la machine.
<i>Joignable (reachable)</i>	Si la machine est joignable par le serveur.
<i>Audit status (audit_status)</i>	Statut de l'audit de tous les paquets de la machine.
<i>WUA (waptwua_status.status)</i>	statut du <i>service WUA</i> via WAPT.
<i>Host (computer_fqdn)</i>	FQDN de l'hôte.
<i>Description (description)</i>	La description de l'hôte
<i>Manufacturer (manufacturer)</i>	Le constructeur de l'hôte
<i>IP addresses (connected_ips)</i>	L'adresse Ip de l'hôte
<i>Plateform (plateform)</i>	Architecture du processeur
<i>Operating system (os_name)</i>	Nom du système d'exploitation
<i>Model (productname)</i>	Modèle de l'hôte
<i>Numéro de série (serialnr)</i>	Numéro de série de la machine
<i>Last seen on (last_seen_on)</i>	Dernière remonté d'inventaire de la machine
<i>Logged in users (last_seen_on)</i>	Utilisateur actuellement connecté
<i>MAC address (mac_addresses)</i>	Adresse MAC correspondantes au IP des machines
<i>WAPT version (wapt_version)</i>	Version de l'agent WAPT déployé sur la machine
<i>uuid (uuid)</i>	Identifiant UUID de la machine
<i>purchase_order (purchase_order)</i>	Numéro de commande d'achat de la machine
<i>purchase_date (purchase_date)</i>	Date d'achat de la machine
<i>Domain controller (host_info/domain_controller)</i>	Nom du contrôleur de domaine de la machine
<i>Groups (depends)</i>	Les groupes dont l'hôte est membre
<i>Domaine (host_info/domain_controller_address)</i>	Nom de domaine
<i>Listening address (listening_address)</i>	Adresse d'écoute
<i>Domain controller address (host_info/domain_controller_address)</i>	Adresse du contrôleur de domaine
<i>AD DN</i>	Distinguished Name de la machine
<i>Last logged on user (last_logged_on_user)</i>	Dernier utilisateur connecté
<i>Registration Method (registration_auth_user)</i>	L'utilisateur qui a enregistré l'hôte sur le serveur WAPT
<i>Repositories (repositories)</i>	Dépôts vus par l'hôte

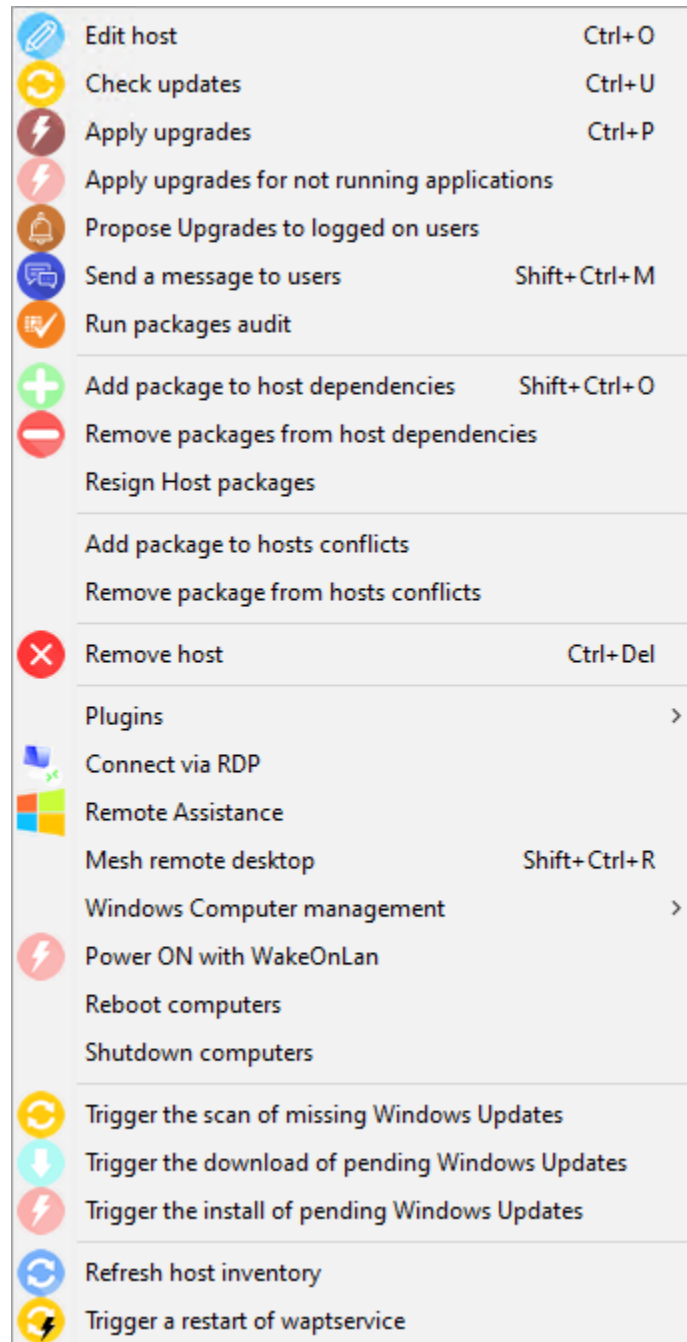
Il est aussi possible de rajouter des colonnes depuis *l'inventaire matériel* en faisant un glissé-déposé.

Vous devez rafraîchir la console WAPT (F5) pour voir les données additionnelles.

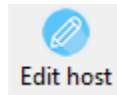
## 64.3 Actions détaillées sur les hôtes

### 64.3.1 Menu de configuration de l'hôte

Les options du menu suivant sont accessibles par un clic-droit sur les postes sélectionnés.



## Modifier la machine



Le bouton *Edit host* vous permet d'éditer la configuration du ou des hôte(s) sélectionné(s).

## Vérification des mises à jours



Le bouton *Check updates* permet de vérifier les mises à jour de la machine sélectionnée.

## Lancer les installations



L'option du menu *Appliquer les mises à jour* permet d'appliquer les mises à jour de la machine sélectionnée.

## Lancer les installations en attentes pour les applications non lancées

L'option du menu *Lancer les installations pour les applications non lancées* vous permet d'appliquer les mises à jour sur le ou les hôte(s) sélectionné(s) **si le logiciel n'est pas actuellement utilisé par un utilisateur.**

## Proposer la mise à jour aux utilisateurs

L'option du menu *Proposer la mise à jour aux utilisateurs* vous permet de proposer des mises à jour aux utilisateurs actuellement connectés sur le ou les hôte(s). **Une fenêtre va alors apparaître sur le terminal permettant à l'utilisateur de déclencher ou reporter la mise à jour proposée.**

Cette option du menu est équivalent à *waptexit*.

## Envoyer un message aux utilisateurs

L'option du menu *Envoyer un message aux utilisateurs* vous permet d'envoyer aux utilisateurs connectés un message sur le ou les hôte(s) sélectionné(s).

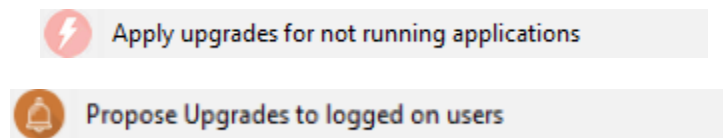


FIG. 1 – Option du menu pour proposer la mise à jour aux utilisateurs

### Lancer l'audit des paquets

L'option du menu *Lancer l'audit des paquets* vous permet de déclencher manuellement un *audit de conformité* sur le ou les hôte(s) sélectionné(s).

### Ajouter un ou des paquet(s) à un ou des hôte(s)

L'option du menu *Ajouter des paquets aux machines* vous permet d'*ajouter une liste de paquets* sur le ou les hôte(s) sélectionné(s).

### Retirer des paquets aux machines

L'option du menu *Retirer des paquets aux machines* vous permet de *supprimer une liste de paquets WAPT* sur un ou des hôte(s) sélectionné(s).

### Resigner les paquets de configuration machine

L'option du menu *Resigner les paquets de configuration machine* vous permet de *resigner la configuration WAPT* du ou des hôte(s) sélectionné(s).

## 64.3.2 Ajouter un paquet en conflit avec l'hôte

L'option du menu *Interdire des paquets* vous permet d'interdire un paquet d'être installé sur le ou les hôte(s) sélectionné(s).

### Autoriser des paquets

L'option du menu *Autoriser des paquets* vous permet de ré-autoriser un paquet interdit sur le ou les hôte(s) sélectionné(s).

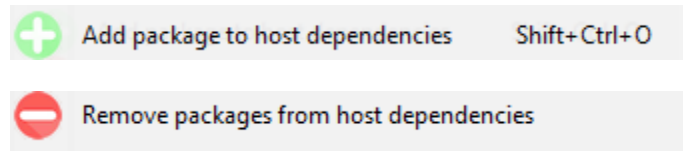
### Supprimer le poste

L'option du menu *Supprimer le poste* vous permet de supprimer le ou les hôte(s) sélectionné(s) depuis la base de données du serveur WAPT.

2 options sont disponibles :

- Effacer la machine de la base de données pour l'effacer des données d'inventaire.
- Effacer la configuration de la machine (Paquet) pour effacer les données du paquet hôte.





## Outils externes

Voir [ici](#).

## Se connecter en RDP

L'option du menu *Se connect en RDP* vous permet de vous connecter avec du RDP sur l'hôte sélectionné.

## Assistance à distance

L'option du menu *Assistance à distance* vous permet de lancer l'Assistance à distance de Windows.

## Bureau à distance Mesh

L'option du menu *Bureau à distance Mesh* permettant l'utilisation de [Mesh Commander](#) via la console WAPT.

## Gestion Windows de l'ordinateur

L'option du menu *Gestion Windows de l'ordinateur* vous permet de lancer l'outils de gestion Windows à distance depuis la console WAPT.


## Force la mise à jour des GPO sur les machines

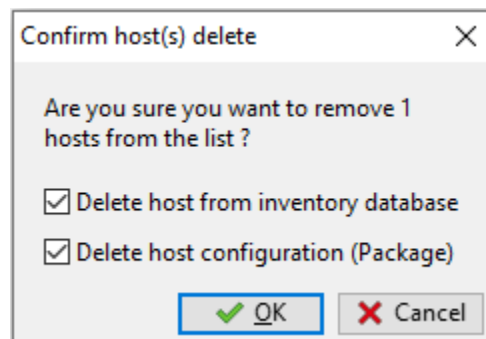
L'option du menu *Forcer la mise à jour des GPO sur les machines* vous permet de lancer la commande `gpupdate /force` sur le ou les hôte(s) sélectionné(s).

## Lance le CleanMgr sur l'hôte

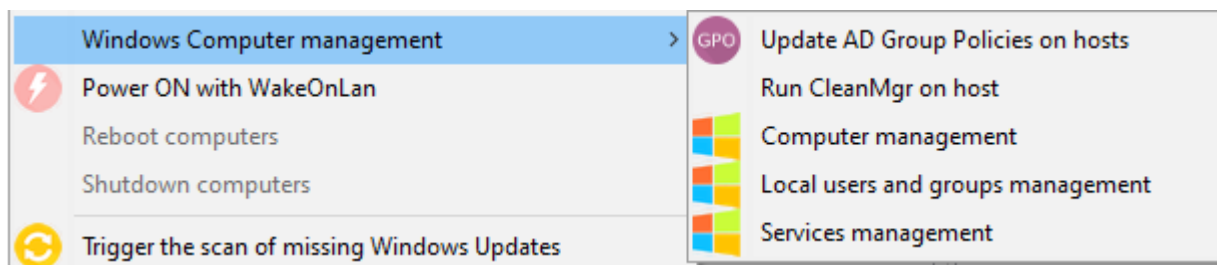
L'option du menu *Lancer CleanMgr sur la machine* vous permet de déclencher l'outil `cleanmgr` à distance sur le ou les hôte(s) sélectionné(s).


A single menu item with the text 'Resign Host packages' in a light gray background.

- Add package to hosts conflicts
- Remove package from hosts conflicts
-  Remove host Ctrl+Del



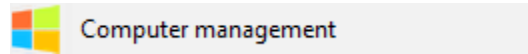
- Plugins >
-  Connect via RDP
-  Remote Assistance



 Update AD Group Policies on hosts

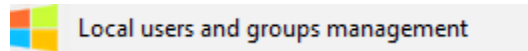
Run CleanMgr on host

## Gérer l'ordinateur



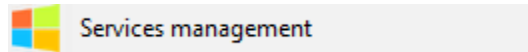
L'option du menu *Gérer l'ordinateur* vous permet de lancer **compmgmt.msc** sur l'hôte sélectionné à distance.

## Gérer les utilisateurs et groupes



L'option du menu *Gérer les utilisateurs et groupes* vous permet de lancer **lusrmgr.exe** sur l'hôte sélectionné à distance.

## Gérer les services



L'option du menu *Gérer les services* vous permet de lancer le **services.msc** sur l'hôte sélectionné à distance.

## Démarre avec le WakeOnLan

L'option du menu *Démarrer avec WakeOnLan* vous permet de réveiller à distance le ou les hôte(s) sélectionné(s) si les paquets magiques sont autorisés.

## Redémarrer les machines

L'option du menu *Redémarrer les machines* vous permet de redémarrer les hôtes sélectionnés à distance.

Cette fonctionnalité est désactivés par défaut, voir *Configurer l'agent WAPT* pour activer cette fonctionnalité.

## Arrêter les machines

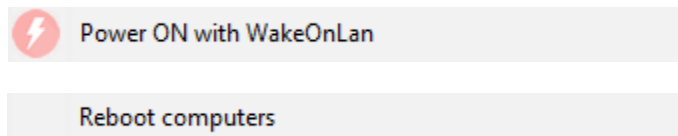
L'option du menu *Eteindre les machines* vous permet d'éteindre les hôtes sélectionnés à distance.

Cette fonctionnalité est désactivés par défaut, voir *Configurer l'agent WAPT* pour activer cette fonctionnalité.

## Lancer la recherche des Mises à jour Windows à appliquer

L'option du menu *Lancer la recherche des mises à jours Windows à appliquer* vous permet de déclencher le scan des Windows Updates manquantes à distance sur le ou les hôte(s) sélectionné(s).

Pour plus d'informations, allez *ici*.



### Lancer le téléchargement des mises à jours Windows en attente

L'option du menu *Lancer le téléchargement des mises à jours Windows en attente* vous permet de déclencher le téléchargement des Windows Updates en attente à distance sur le ou les hôte(s) sélectionné(s).

Pour plus d'informations, allez *ici*.

### Lancer l'installation des mises à jours Windows en attente

L'option du menu *Lancer l'installation" des mises à jours Windows en attente* vous permet de déclencher l'installation des Windows Updates en attente à distance sur le ou les hôte(s) sélectionné(s).

Pour plus d'informations, allez *ici*.

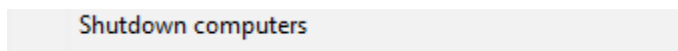
### Actualiser l'inventaire des postes

L'option du menu *Actualiser l'inventaire des postes* vous permet de forcer le ou les hôte(s) sélectionné(s) à renvoyer leur inventaire actuel.





Colonne	Description
Hôte	Nom de la machine
IP	IP de la machine
Status	Statut de l'action
status	Statut visuel
Message	Message suite à la demande de mise à jour
<i>Launch update</i>	Lancer l'action
<i>Fermer</i>	Ferme la fenêtre

### Lancer le redémarrage de waptservice

L'option du menu *Lancer le redémarrage waptservice* vous permet de redémarrer de force le service **waptservice** sur le ou les hôte(s)sélectionné(s).







-  Trigger the scan of missing Windows Updates
-  Trigger the download of pending Windows Updates
-  Trigger the install of pending Windows Updates
-  Refresh host inventory

Trigger packages list update on the hosts

Host	IP	Status	st...	Message
wsmange-demo.test.tranquil.it	e81155124d0c4f8591646e0b43aa3183			

Terminate Launch update  Close

-  Trigger a restart of waptservice

## 64.4 Sous onglet de l'hôte

Overview Hardware inventory Software inventory Windows updates Tasks Packages overview

Name	<input type="text"/>	Manufacturer	<input type="text"/>
Description	<input type="text"/>	Model	<input type="text"/>
Operating system	<input type="text"/>	Last seen	<input type="text"/>
IP address	<input type="text"/>	Logged in users	<input type="text"/>
Last task	<input type="text"/>	UUID	<input type="text"/>

Search keywords    Errors   To upgrade   To install   To remove All Pending

Statu	Audit status	Package name	Version	Install date	Section	Architecture	Locale
.....							

<  >

>

.....

^

v

## 64.4.1 Onglet Général

Overview Hardware inventory Software inventory Windows updates Tasks Packages overview

Name	WSMANAGE-DEMO	Manufacturer	Xen
Description		Model	HVM domU
Operating system	Windows 10 Pro	Last seen	2021-08-23 08:53
IP address	192.168.154.156,fe80::5067:bae8:c0ca:	Logged in users	tisadmin
Last task	Done: Mise à jour de la liste des paqu	UUID	6730C5A2-2CF8-47FF-76BC-D697E116

Search keywords

Errors 
  To upgrade 
  To install 
  To remove

Statu	Audit status	Package name	Version	Install date	Section	Architecture	Locale
✓	✓ OK	test-rsat-w10-2004	1.0-6	2021-04-30 16:11	base	x64	all
✓	✓ OK	test	1	2021-06-03 10:54	selfse...	all	
✓	✓ OK	bat1-bid	1	2021-06-03 10:54	group	all	
✓	✓ OK	OU=tranquilit_DC=ad_DC=tran...	1	2021-06-03 10:54	unit	all	
✓	✓ OK	test-vlc	3.0.16-7	2021-08-20 11:36	base	x64	all
✓	✓ OK	test-firefox-esr	78.13.0-107	2021-08-20 15:12	base	x64	fr
✓	✓ OK	6730C5A2-2CF8-47FF-76BC-D69...	10	2021-08-20 15:12	host	all	

<  >

> Total : 7 elements

.....

## 64.5 Host informations

Overview	Hardware inventory	Software inventory	Windows updates	Tasks	Packages overview
Name	WSMANAGE-DEMO			Manufacturer	Xen
Description				Model	HVM domU
Operating system	Windows 10 Pro			Last seen	2021-08-23 08:53
IP address	192.168.154.156,fe80::5067:bae8:c0ca:			Logged in users	tisadmin
Last task	Done: Mise à jour de la liste des paqu			UUID	6730C5A2-2CF8-47FF-76BC-D697E110

TABLEAU 1 – Les informations affichées dans l’onglet *Général*

<i>Name</i>	Le nom de l’hôte
<i>Description</i>	La description de l’hôte
<i>Operating system</i>	Le système d’exploitation tournant sur l’hôte
<i>IP address</i>	L’adresse Ip de l’hôte
<i>Last task</i>	La dernière tâche WAPT qui s’est effectuée sur l’hôte
<i>Manufacturer</i>	Le constructeur de l’hôte
<i>Model</i>	Le modèle de l’hôte
<i>Last seen</i>	La date à laquelle la dernière mise à jour de l’hôte s’est passée
<i>Logged in users</i>	Le nom du dernier utilisateur connecté ou actuellement connecté sur l’hôte
<i>UUID</i>	L’UUID BIOS de l’hôte

## 64.6 Mots-clés de recherche

## 64.7 Filtre des statut de paquet

Errors
  To upgrade
  To install
  To remove

Les Cases permettent de filtrer les paquets par leur *statut*.

TABLEAU 2 – Statut des paquets dans la console WAPT

Status	Description
<i>Errors</i>	Liste des paquets qui ne se sont pas installé correctement.
<i>To upgrade</i>	Liste des paquets dont une mise à jour est disponible.
<i>To install</i>	Liste des paquets en attente d’installation.
<i>To remove</i>	Liste des paquets en attente de désinstallation.

Le bouton *All* permet de sélectionner tous les états.

Le bouton *Pending* permet de sélectionner les paquets en statut :

- *NEED-UPGRADE* ;
- *NEED-INSTALL* ;
- *TO-REMOVE*.

## 64.8 Affichage du paquet

Status	Audit status	Package name	Versi ▲	Install date	Section	Architecture	Locale	Maturity	Audited on	Next audit
--------	--------------	--------------	---------	--------------	---------	--------------	--------	----------	------------	------------

Il est possible de sélectionner plus de colonnes avec un Clic-Droit sur l'en-tête de la grille.

TABLEAU 3: Statut des paquets dans la console WAPT

	Status	Description
<i>Status (install_status)</i>		Statut du paquet WAPT.
<i>Audit status (last_audit_status)</i>		Dernière date de l'audit de conformité.
<i>Version (version)</i>		Version du paquet WAPT.
<i>Description (description)</i>		La description du paquet WAPT.
<i>Dependencies (depends)</i>		Dépendances du paquet WAPT.
<i>installé le (install_date)</i>		Date d'installation du paquet WAPT.
<i>priority</i>		<b>Pas utilisé.</b>
<i>sources (sources)</i>		L'URL pour télécharger le logiciel original.
<i>Installé par (explicit_by)</i>		L'utilisateur qui a installé le paquet sur l'hôte.
<i>Section (section)</i>		Type de paquet WAPT.
<i>Architecture (architecture)</i>		Architecture CPU du paquet.
<i>Locale (locale)</i>		Langue du paquet WAPT.
<i>md5sum (md5sum)</i>		somme md5 du paquet WAPT.
<i>File name (filename)</i>		Nom du paquet sur le dépôt WAPT.
<i>Parameters (install_params)</i>		Liste des paramètres d'installation du paquet WAPT.
<i>Log install (install_output)</i>		Logs d'installation du paquet WAPT.
<i>maintainer (maintainer)</i>		Nom du mainteneur du paquet WAPT.
<i>conflict (conflict)</i>		Liste de paquet interdits. (paquets qui doivent être désinstallés avant que le paque ne s'installe).
<i>Taille après installation (installed_size)</i>		Espace occupé par le logiciel une fois installé.
<i>URL du dépôt (repo_url)</i>		URL du dépôt duquel le paquet a été téléchargé.
<i>Taille (size)</i>		Taille du paquet WAPT dans le dépôt.
<i>Nom du dépôt (repo_url)</i>		FQDN du dépôt duquel la paquet a été téléchargé.
<i>Maturité (maturity)</i>		Niveau de maturité du paquet WAPT ( <b>DEV</b> , <b>PREPROD</b> , <b>PROD</b> ).
<i>Audité le</i>		Date du dernier audit.
<i>Next audit (next_audit_on)</i>		Date du prochain audit de conformité.
<i>clé de désinstallation (uninstall_key)</i>		Clé de désinstallation du logiciel.
<i>update_on (update_on)</i>		Date de la dernière mise à jour de paquet.
<i>update_by (update_by)</i>		Utilisateur ayant la dernière mise à jour du paquet.
<i>created_on (created_on)</i>		Date de la première installation du paquet WAPT.
<i>created_by (created_by)</i>		Utilisateur ayant installé le paquet WAPT pour la première fois.
<i>last_audit_output (last_audit_output)</i>		Log du dernier audit de conformité
<i>id (id)</i>		ID du paquet dans la base de donnée <code>sqlite</code> locale de l'agent WAPT.

Status (install\_status)  
Audit status (last\_audit\_status)  
Package name (package)  
Version (version)  
Description (description)  
Dependencies (depends)  
Install date (install\_date)  
priority (priority)  
sources (sources)  
Installed by (explicit\_by)  
Section (section)  
Architecture (architecture)  
Locale (locale)  
md5sum (md5sum)  
File name (filename)  
Parameters (install\_params)  
Log install (install\_output)  
maintainer (maintainer)  
conflicts (conflicts)  
Size after install (installed\_size)  
URL of repository (repo\_url)  
Size (size)  
Name of repository (repo)  
localpath (localpath)  
Maturity (maturity)  
Audited on (last\_audit\_on)  
Next audit (next\_audit\_on)  
uninstall\_key (uninstall\_key)  
updated\_on (updated\_on)  
updated\_by (updated\_by)  
created\_on (created\_on)  
created\_by (created\_by)  
last\_audit\_output (last\_audit\_output)  
id (id)  
signer\_fingerprint (signer\_fingerprint)  
host (host)  
signer (signer)  
Package UUID (package\_uuid)  
Install ID (install\_id)  
Forced install on (forced\_install\_on)

## 64.9 Statut du paquet

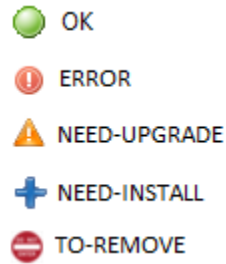


TABLEAU 4 – Statut des paquets dans la console WAPT

Status	Description
<i>OK</i>	Paquet installé
<i>ERROR</i>	Paquet ayant subit une erreur d'installation
<i>NEED-UPGRADE</i>	Paquet ayant une mise à jour en attente
<i>NEED-INSTALL</i>	Paquet en attente d'installation
<i>TO-REMOVE</i>	Paquet en attente d'installation

Il est possible d'avoir des informations sur les status en cliquant dessus.

Par exemple, ici un statut d'erreur :

Status	Audit status	Package name
<span style="color: red;">! ○</span> ERROR		test-firefox-esr
<span style="color: blue;">+</span> NEED-INSTALL		test-firefox-esr
<span style="color: green;">✓</span> OK	<span style="color: green;">✓</span> OK	test-rsat-w10-2004
<span style="color: green;">✓</span> OK	<span style="color: green;">✓</span> OK	test-vlc

Selected / Total : 1 / 8

---

**Install logs of package test-firefox-esr**

```

Installing: Firefox_Setup_78.13.0esr.exe
Traceback (most recent call last):
  File "C:\Program Files (x86)\wapt\common.py", line 3923, in install_wapt
    exitstatus = setup.install()
  File "C:\WINDOWS\TEMP\waptimupnmr7\setup.py", line 39, in install
  File "C:\Program Files (x86)\wapt\common.py", line 3877, in new_func
    return func(*args, **kwargs)
  File "C:\Program Files (x86)\wapt\setuptools_helpers_windows.py", line 1335, in install
    error('setup exe file %s not found in package' % exe)
  File "C:\Program Files (x86)\wapt\waptutils.py", line 1841, in error

```

## 64.10 Agir sur les paquets installés sur un hôte

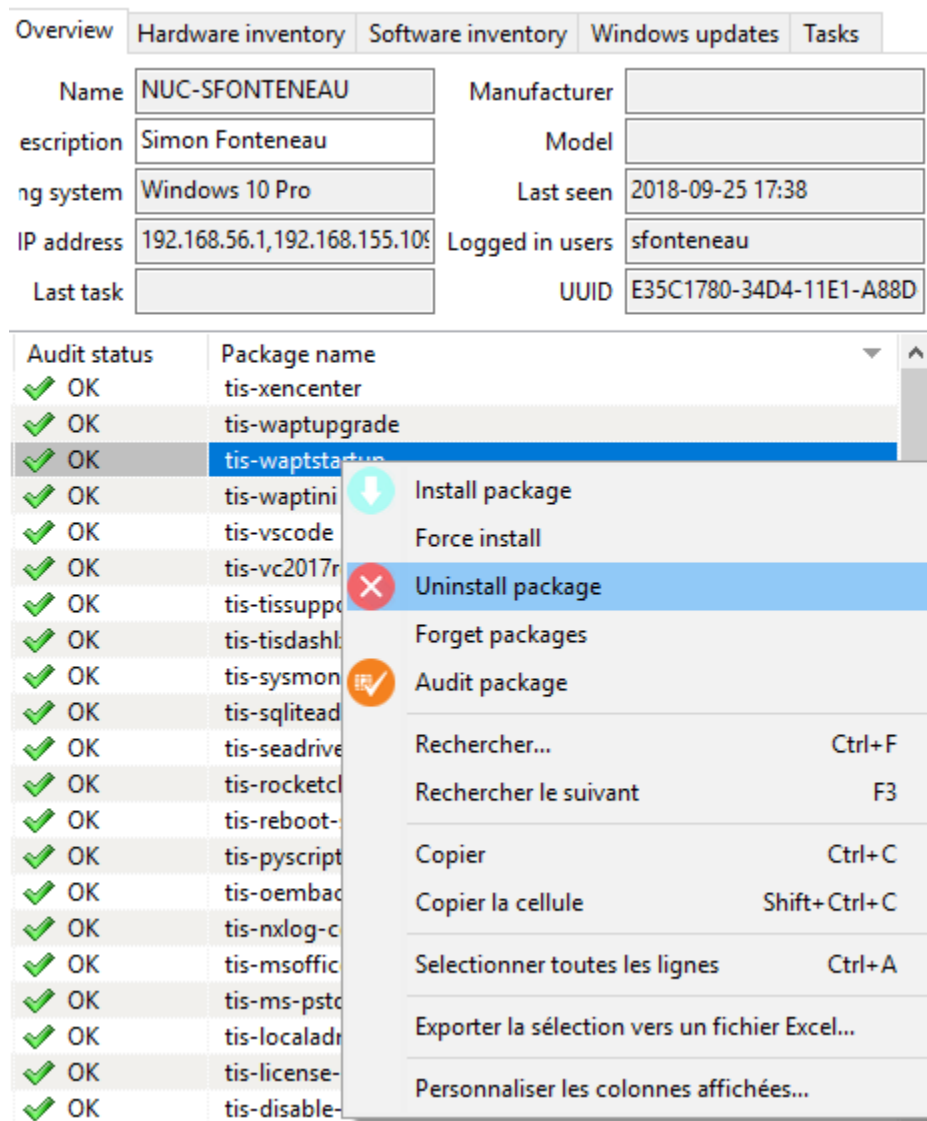


FIG. 2 – Actions possibles sur les paquets WAPT

### Indication :

- Sélectionner plusieurs paquets si possible ;
- Si plusieurs hôtes sont sélectionnés, l'action se lancera sur tous les hôtes sélectionnés ;



TABLEAU 5 – Paramètres disponibles pour la section [option] du waptserver.ini

Action	Description
Installer un paquet	L'action install le paquet sélectionné sur le ou les hôte(s) sélectionné(s).
Forcer un paquet	L'action force la ré-installation d'un paquet sélectionné sur le ou les hôte(s) sélectionné(s).
Supprimer un paquet	L'action supprime le paquet sélectionné du/des hôte(s) sélectionné(s)
Oublier un paquet	L'action dit que le ou les hôte(s) sélectionné(s) <b>n'utilisent plus WAPT</b> pour gérer les logiciels ou la configuration
L'audit de paquet <small>de wapt</small>	L'action déclenche un audit sur le ou les paquet(s) sélectionné(s)

## 64.11 Onglet Inventaire Matériel

Les informations affichées par défaut dans l'*Inventaire Matériel* sont :

- L'information des composants matériel de l'hôte.
- Des informations à propos de l'hôte.
- Des informations à propos du statut de WAPT.

Une case *Filtre* permet de rechercher des hôtes.

---

**Indication :** Les filtres fonctionnent avec les *expressions régulières*.

---

Pour ajouter une colonne dans la grille, glissez et déposez la propriété du matériel depuis la liste de l'*inventaire Matériel* vers la grille principale.

Exemple : dans *hosts*, glissez et déposez *physical\_memory* dans le panneau de gauche, et la colonne *physical\_memory* va apparaître dans la grille principale.

## 64.12 Onglet inventaire Logiciel

Les informations communes affichées dans l'onglet de l'*Inventaire Logiciel* sont :

- *maker* ;
- *software name* ;
- *version* ;
- *installation date* ;
- *clé de désinstallation* ;
- *uninstall string*.

Overview Hardware inventory Software inventory Tasks

Filter:  Add item as grid column

Property	Value
⊕ wmi	
⊕ dmi	
⊖ host_info	
⊕ profiles_users	
⊕ local_administrators	
⊖ mac	
0	42:c3:40:63:7f:c7
system_productname	HVM domU
⊖ connected_ips	
0	192.168.149.149
⊖ local_drives	
⊕ D	
⊕ C	
domain_name	null
⊖ current_user	
0	admin
domain_controller	null
wua_agent_version	7.6.7601.23806
virtual_memory	2147352576
computer_ad_site	
⊖ windows_startup_items	
run	
⊕ common_startup	
system_manufacturer	Xen
description	administrateur demo
computer_ad_dn	
registered_organization	Orgname
win64	True
⊖ networking	
⊕ 0	
domain_controller_address	null
⊖ windows product infos	

The screenshot displays the WAPT console interface. At the top, there is a menu bar (File, Actions, View, Tools, ?) and a toolbar with icons for Refresh, Edit host, Check updates, and Apply upgrades. Below the toolbar, there are search and filter options, including a search keywords field, a search refresh button, and checkboxes for 'More options', 'Has errors', 'Needs updating', and 'Connected only'. There are also dropdown menus for 'Groups' and 'Site'.

The main area is divided into two panes. The left pane shows a table of host inventory:

S...	Host	Description	IP addresses	Windows version	Model
✓	wsmanage-demo	administrateur demo	192.168.149.149	Windows 7 Professional	HVM domU
✓	wsclient7-demo	client demo	192.168.149.169	Windows 7 Professional	HVM domU

The right pane shows a detailed view of the selected host's properties, organized into a tree structure. The 'wua\_agent\_version' property is highlighted in blue. The properties and their values are:

- wmi
- dmi
- host\_info
  - profiles\_users
  - local\_administrators
  - mac
    - 0: 42:c3:40:63:7f:c7
  - system\_productname: HVM domU
- connected\_ips
  - 0: 192.168.149.149
- local\_drives
  - D
  - C
- domain\_name: null
- current\_user
  - 0: admin
- domain\_controller: null
- wua\_agent\_version: 7.6.7601.23806
- virtual\_memory: 2147352576
- computer\_ad\_site
- windows\_startup\_items
  - run
    - common\_startup
  - system\_manufacturer: Xen
  - description: administrateur demo
  - computer\_ad\_dn
  - registered\_organization: Orgname
  - win64: True
- networking
  - 0
  - domain\_controller\_address: null
  - windows\_product\_infos

At the bottom of the console, there is a status bar showing the user 'admin on waptconsole 1.5.1.19 WAPT Enterprise Edition, (c) 2012-2017 Tranquil IT Systems. (Configuration:C:\Users\agauvrit-adm\AppData\Local\waptconsole\waptconsole.ini)

Overview Hardware inventory Software inventory Tasks

Filter :   Hide system components

Software name ▲	Version	Install date	Publisher	Uninstall key
BG Info	4.20	20171020	SysInternals	bginfo
Citrix Xen Windows x64 PV Drivers	6.5.138	20160107	Citrix	{74C6D7F6-76A2-444C-8F8..
Citrix XenServer Tools Installer	6.5.138	20160107	Citrix	{BE822B8D-09AF-4048-953...
Citrix XenServer Windows Guest Agent	6.5.138	20160107	Citrix	{A8ACDDFC-777B-46EA-A...
Microsoft .NET Framework 4.7	4.7.02053		Microsoft Corp...	{92FB6C44-E685-45AD-9B2..
Microsoft .NET Framework 4.7 (Français)	4.7.02053		Microsoft Corp...	{92FB6C44-E685-45AD-9B2..
Microsoft Visual C++ 2008 Redistributable - x86 9.0.210...	9.0.21022	20130408	Microsoft Corp...	{FF66E9F6-83E7-3A3E-AF14..
Microsoft Visual C++ 2008 Redistributable - x86 9.0.307...	9.0.30729.6...	20160415	Microsoft Corp...	{9BE518E6-ECC6-35A9-88E...
Mozilla Firefox 52.6.0 ESR (x86 fr)	52.6.0		Mozilla	Mozilla Firefox 52.6.0 ESR (x..
Mozilla Maintenance Service	52.6.0		Mozilla	MozillaMaintenanceService
Notepad++ (64-bit x64)	7.5.5		Notepad++ Team	Notepad++
Package de pilotes Windows - Citrix Systems Inc. (xenb...	06/12/2014...		Citrix Systems Inc.	67C40D08D848E005122CD1..
Package de pilotes Windows - Citrix Systems Inc. (xenn...	04/15/2014...		Citrix Systems Inc.	7DB29F50EFDC3E3B04C5B...
Package de pilotes Windows - Citrix Systems Inc. (xenv...	11/03/2014...		Citrix Systems Inc.	885B57592CE8A2FA9977C4...
Package de pilotes Windows - Citrix Systems, Inc. (xeni...	03/25/2014...		Citrix Systems, I...	C13191ADAF0BA398DFDBB..
Package de pilotes Windows - Citrix Systems, Inc. (xen...	06/20/2014...		Citrix Systems, I...	CAB8C46CF641BAD973C19..
Update for Microsoft .NET Framework 4.7 (KB4040973)	1		Microsoft Corp...	{92FB6C44-E685-45AD-9B2..
Update for Microsoft .NET Framework 4.7 (KB4043764)	1		Microsoft Corp...	{92FB6C44-E685-45AD-9B2..
VLC media player	3.0.1		VideoLAN	VLC media player
WAPTagent 1.5.1.19	1.5.1.19	20180309	wapt-private	WAPT_is1

## 64.13 L'onglet Windows update WAPT

Les informations affichées dans l'onglet *Windows update* sont :

- La version Windows update de l'agent.
  - Date du dernier scan Windows update.
  - Durée du dernier scan.
  - Statut WAPTWUA.
  - Date de la dernière version du `wsusscn2.cab` fourni par WAPT.
  - Statut de l'activation de WAPTWUA (True/ False).
- La grille listes alors les fichiers cab Windows qui ont été installé ou qui sont en attente d'installation.

Les informations affichées dans l'onglet *Windows update* sont :

- *Statut* ;
- *Produit* ;
- *ID de la mise à jour* ;
- *KB ids* ;
- *Publié le* ;
- *installé le* ;
- *Criticité* ;
- *Classification* ;
- *Produit* ;
- *Taille du Téléchargement*.

Overview
Hardware inventory
Software inventory
Windows updates
Tasks

<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">WUA Status</td> <td>PENDING_UPDATES</td> </tr> <tr> <td>WSUS Scan Cab Date</td> <td>2018-09-11T11:57:04</td> </tr> <tr> <td>WAPT WUA Enabled</td> <td>true</td> </tr> </table>	WUA Status	PENDING_UPDATES	WSUS Scan Cab Date	2018-09-11T11:57:04	WAPT WUA Enabled	true	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Windows Agent version</td> <td>10.0.17134.280</td> </tr> <tr> <td>Last scan date</td> <td>2018-09-17T15:02:10.310000</td> </tr> <tr> <td>Last scan duration</td> <td>161</td> </tr> </table>	Windows Agent version	10.0.17134.280	Last scan date	2018-09-17T15:02:10.310000	Last scan duration	161
WUA Status	PENDING_UPDATES												
WSUS Scan Cab Date	2018-09-11T11:57:04												
WAPT WUA Enabled	true												
Windows Agent version	10.0.17134.280												
Last scan date	2018-09-17T15:02:10.310000												
Last scan duration	161												

Critical only   
  Installed   
  Pending   
  Discarded

Title	Classification	Product	Up...	kbids	Published on
Mise à jour de sécurité pour Microsoft Office 2010 (KB2956076) É...	Security Updates	Office 2010	f3201f...	KB2956076	2015-03-10 00:00:00
2018-09 Mise à jour cumulative pour Windows 10 Version 1803 p...	Security Updates	Windows 10	eab1a4...	KB4457128	2018-09-11 00:00:00
Service Pack 1 pour le moteur de base de données Microsoft Acc...	Service Packs	Office 2010	da274c...	KB2460011	2011-06-28 00:00:00
Mise à jour de sécurité pour Microsoft Office 2010 (KB4022198) É...	Security Updates	Office 2010	cc4958...	KB4022198	2018-08-14 00:00:00
Mise à jour de sécurité pour le package redistribuable Microsoft V...	Security Updates	Visual Studio 2005	bb49cc...	KB2538242	2012-01-24 00:00:00
Mise à jour de sécurité pour Microsoft Office 2010 (KB4022206) É...	Security Updates	Office 2010	bac66a...	KB4022206	2018-07-10 00:00:00
Mise à jour de sécurité pour Microsoft Office 2010 (KB3114874) É...	Security Updates	Office 2010	9fae99...	KB3114874	2018-02-13 00:00:00
Outil de suppression de logiciels malveillants Windows x64 - sept...	Update Rollups	Windows 10	35b02a...	KB890830	2018-09-11 00:00:00
2018-09 Mise à jour de sécurité pour Adobe Flash Player sous Win...	Security Updates	Windows 10	1f0e47...	KB4457146	2018-09-11 00:00:00
Mise à jour de sécurité de MSXML 6.0 RTM (925673)	Security Updates	SQL Server Feature Pack	07609d...	KB925673	2012-04-04 00:00:00

## 64.14 L'onglet tâche

Les informations affichées par défaut dans l'onglet *Tâches* sont :

- Tâches en attente.
- Tâches terminées.
- Tâches en erreur.

WAPT console

File Actions View Tools ?

Inventory Bundles Softwares repository

Refresh Edit host Check updates Apply upgrades

Search keywords Search Refresh More options Has errors Groups

Not Needs updating Site

Search in All Host Hardware Software Packages Connected only

S...	Host	Description	IP addresses	Windows versior
wsmanage-demo	administrateur demo	192.168.149.149	Windows 7 Prof	
wsclient7-demo	client demo	192.168.149.169	Windows 7 Prof	

Overview Hardware inventory Software inventory Tasks

Current task

Installing test-adobereader

Cancel current task

Pending Done Errors

description	Created on	Created ...
6 Vider le cache de paquets	2018-03-09T17...	wapt-pr...
5 Mettre à jour les paquets installés sur le poste	2018-03-09T17...	wapt-pr...
4 Installation de 0AFBC5F4-46A8-3AC1-AE22-C71F5CA...	2018-03-09T17...	wapt-pr...

WAPT console

File Actions View Tools ?

Inventory Bundles Softwares repository

Refresh Edit host Check updates Apply upgrades

Search keywords Search Refresh More options Has errors Groups

Not Needs updating Site

Search in All Host Hardware Software Packages Connected only

S...	Host	Description	IP addresses	Windows versior
wsmanage-demo	administrateur demo	192.168.149.149	Windows 7 Prof	
wsclient7-demo	client demo	192.168.149.169	Windows 7 Prof	

Overview Hardware inventory Software inventory Tasks

Current task

Cancel current task

Pending Done Errors

description	summary	Finished on	Created by
6 Vider le cache de paquets	Paquets effacés : C:\Program Files (x86)...	2018-03-09T17...	wapt-private
5 Mettre à jour les paquets installés sur le poste	Le système est à jour	2018-03-09T17...	wapt-private
4 Installation de 0AFBC5F4-46A8-3AC1-AE22-C71F5CA...	Installé : 0AFBC5F4-46A8-3AC1-AE22-C...	2018-03-09T17...	wapt-private
3 Installation de test-adobereader (=2018.11.20035-4)...	Installé : test-adobereader (=2018.11.20...	2018-03-09T17...	wapt-private
2 Mise à jour de la liste des paquets disponibles	1 nouveau(x) paquet(s)5 paquet(s) da...	2018-03-09T17...	wapt-private
1 Mise à jour de la liste des paquets disponibles	4 paquet(s) dans le dépôtLe système es...	2018-03-09T17...	SCHEDULER

Overview Hardware inventory Software inventory Windows updates **Tasks** Packages overview

Current task

Cancel current task

Search keywords   Done  Pending  Errors

ID	Status	Description	Summary	Finished on	Cre
10	Error	Installation de test-java(=80-1) (tâche...	Exception: Erreur lors de l'installation d...	2021-03-30T17...	
11	Error	Installation de 23898926-0FE2-F48E-3...	Exception: Erreur lors de l'installation d...	2021-03-30T17...	
12	Error	Audit de test-java(=80-1),23898926-0...	Exception: Package test-java(=80-1) is ...	2021-03-30T17...	

Logs of task Installation de test-java(=80-1) (tâche #10) (ID 10)

Installing test-java(=80-1)

Traceback (most recent call last):  
 File "C:\Program Files (x86)\wapt\waptservice\service.py", line 1958, in run  
 self.running\_task.run()  
 File "C:\Program Files (x86)\wapt\waptservice\waptservice\_common.py", line 649, in run  
 self.\_run()  
 File "C:\Program Files (x86)\wapt\waptservice\waptservice\_common.py", line 1192, in \_run  
 raise Exception(\_('Error during install of {}: errors in packages {}').format(  
 Exception: Erreur lors de l'installation de ['test-java(=80-1)']: erreurs dans les paquets [[PackageRequest(package='test-  
 java',version=(Version('80.0.0.0'), 1),architectures=['x64'],locales=['fr'],maturities=['PROD'],tags=['windows-10',  
 'windows'],min\_os\_version=Version('10.0.18362'),max\_os\_version=Version('10.0.18362')), PackageEntry('test-java','80-1'  
 maturity='PROD',target\_os='windows'), 'Traceback (most recent call last):\n File "C:\\Program Files (x86)\\wapt\\common.py",  
 line 4998, in install\n result = self.install\_wapt(full\_fname(p.filename),\n File "C:\\Program Files (x86)\\wapt\\common.py",  
 line 3997, in install\_wapt\n raise e\n File "C:\\Program Files (x86)\\wapt\\common.py", line 3766, in install\_wapt\n raise  
 EWaptUnavailablePackage(\nMissing dependencies: %s\ ' % (\,'.join  
 (missing\_depends,))\nwaptpackage.EWaptUnavailablePackage: Missing dependencies: test-java8\n']

Exception: Erreur lors de l'installation de ['test-java(=80-1)']: erreurs dans les paquets [[PackageRequest(package='test-  
 java',version=(Version('80.0.0.0'), 1),architectures=['x64'],locales=['fr'],maturities=['PROD'],tags=['windows-10',  
 'windows'],min\_os\_version=Version('10.0.18362'),max\_os\_version=Version('10.0.18362')), PackageEntry('test-java','80-1'  
 maturity='PROD',target\_os='windows'), 'Traceback (most recent call last):\n File "C:\\Program Files (x86)\\wapt\\common.py",  
 line 4998, in install\n result = self.install\_wapt(full\_fname(p.filename),\n File "C:\\Program Files (x86)\\wapt\\common.py",  
 line 3997, in install\_wapt\n raise e\n File "C:\\Program Files (x86)\\wapt\\common.py", line 3766, in install\_wapt\n raise  
 EWaptUnavailablePackaqe(\nMissing dependencies: %s\ ' % (\,'.join

> Selected / Total : 1 / 3

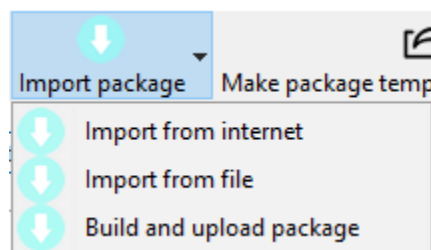
## 64.15 Vue global des paquets

## 64.16 Données d'audit



### 65.1 Rafraîchir la liste des paquets

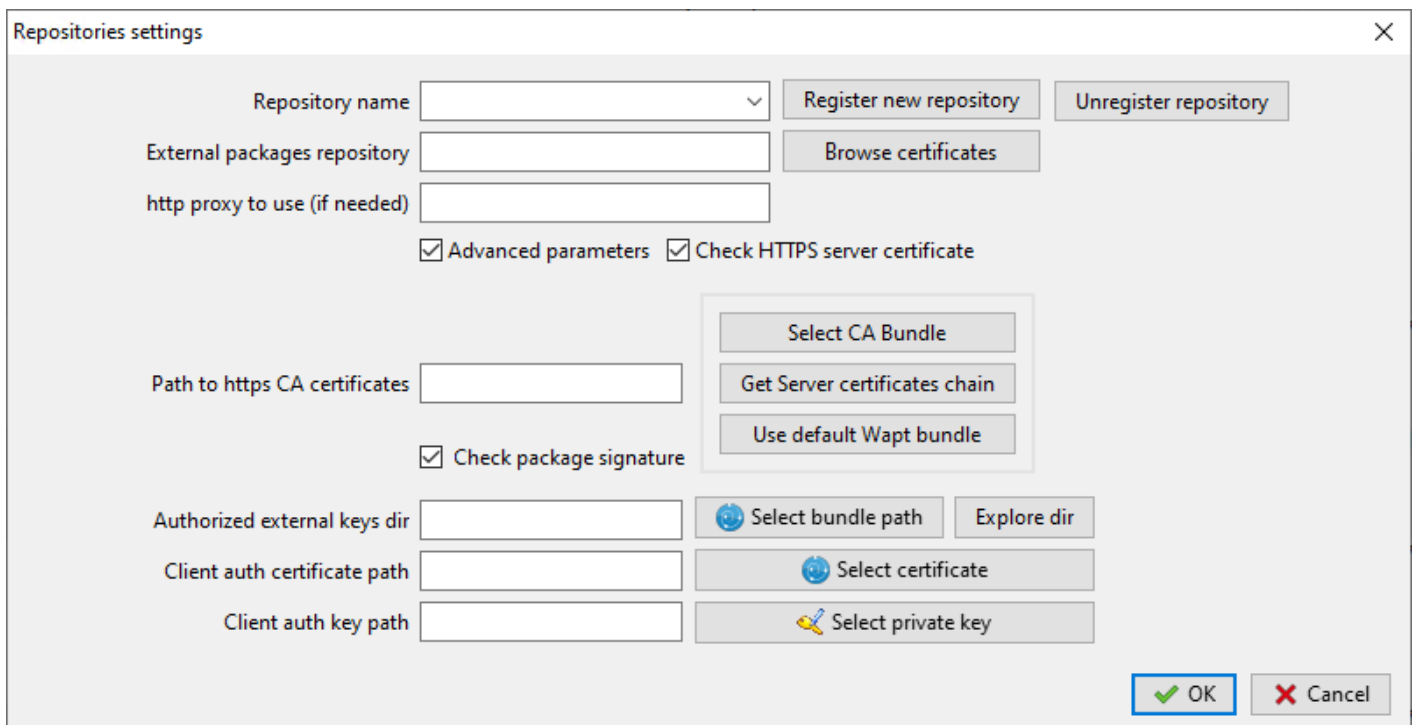
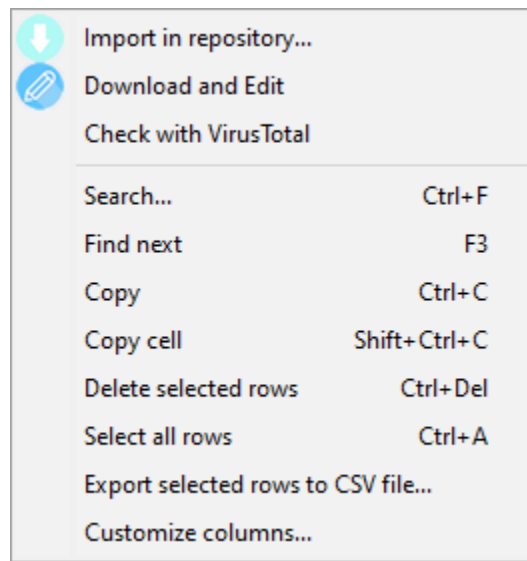
### 65.2 Importer des paquets



**Attention :** Si la vérification de la *signature de paquet* est activée pour une sécurité renforcée, le certificat public du ou des signataire(s) dont les paquets sont stockés dans le dépôt doit être situé dans l'un des dossiers suivant :

- C:\Program Files (x86)\wapt\ssl;
- %appdata%\waptconsole\ssl;

Si le certificat n'est pas trouvé dans un de ces deux dossiers, alors l'erreur suivante va apparaître et aucun paquet ne sera importé depuis le dépôt.



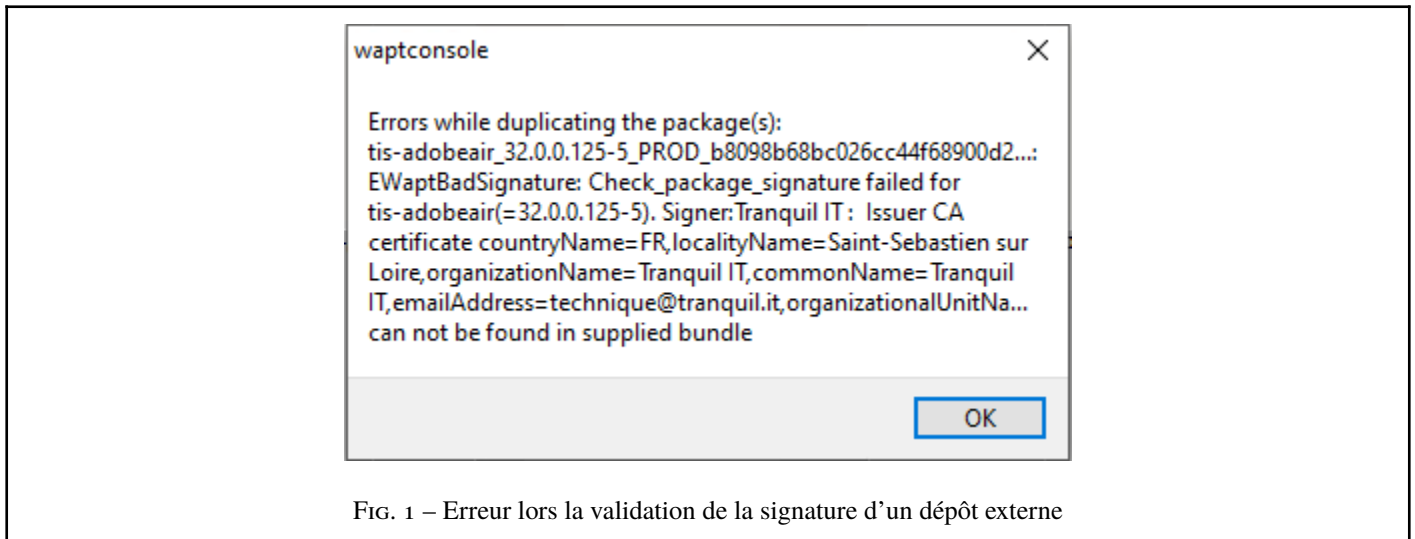
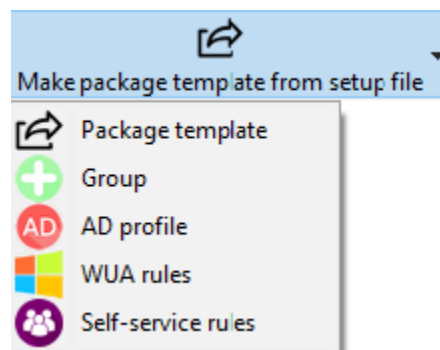


FIG. 1 – Erreur lors la validation de la signature d'un dépôt externe

## 65.3 Importer depuis un fichier

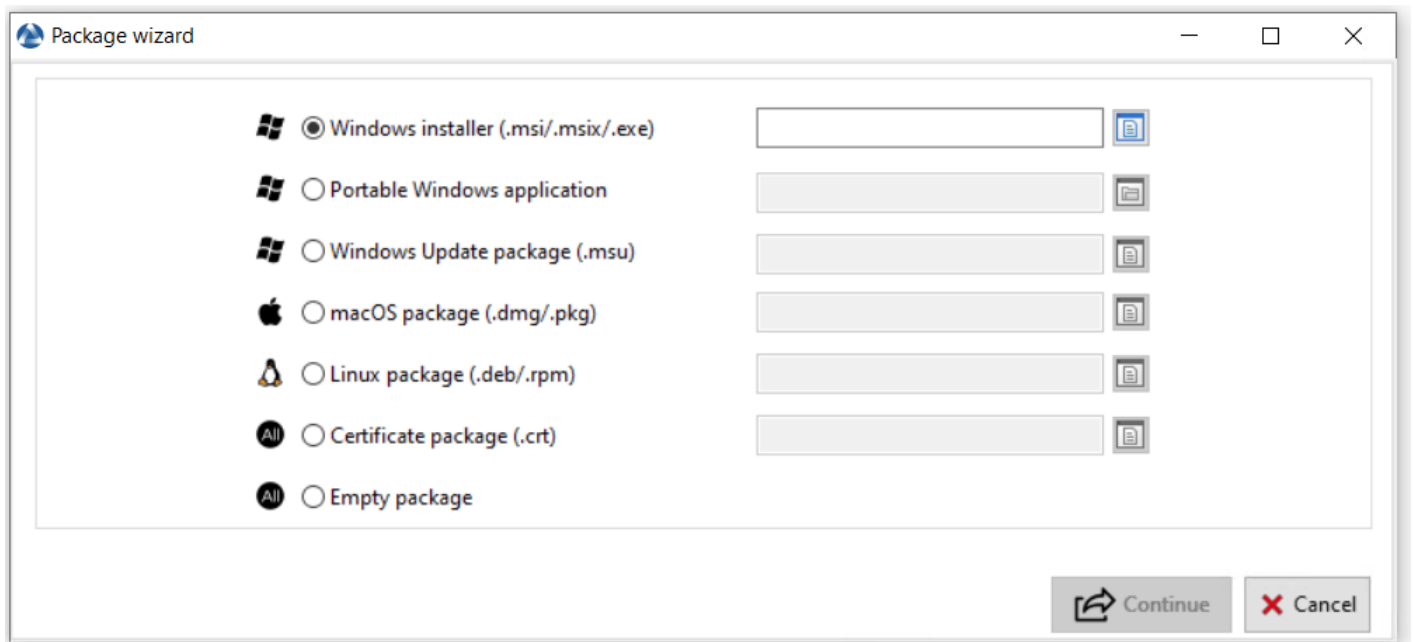
### 65.3.1 Construire et téléverser les paquets

## 65.4 Créer des modèles de paquets à partir d'un fichier setup



### 65.4.1 Modèle de paquet

Pour sélectionner un fichier, cliquer sur :



**Installeur Windows (.msi/.msix/.exe)**

**Application portable Windows**

**Paquet Windows Update (.msu)**

**paquet macOS (.dmg/.pkg)**

**Paquet Linux (.deb/rpm)**

**Paquet de certificate (.crt)**

Sélectionnez .crt puis cliquez sur Continuer

Confirmer les informations puis cliquez sur Créer et téléverser

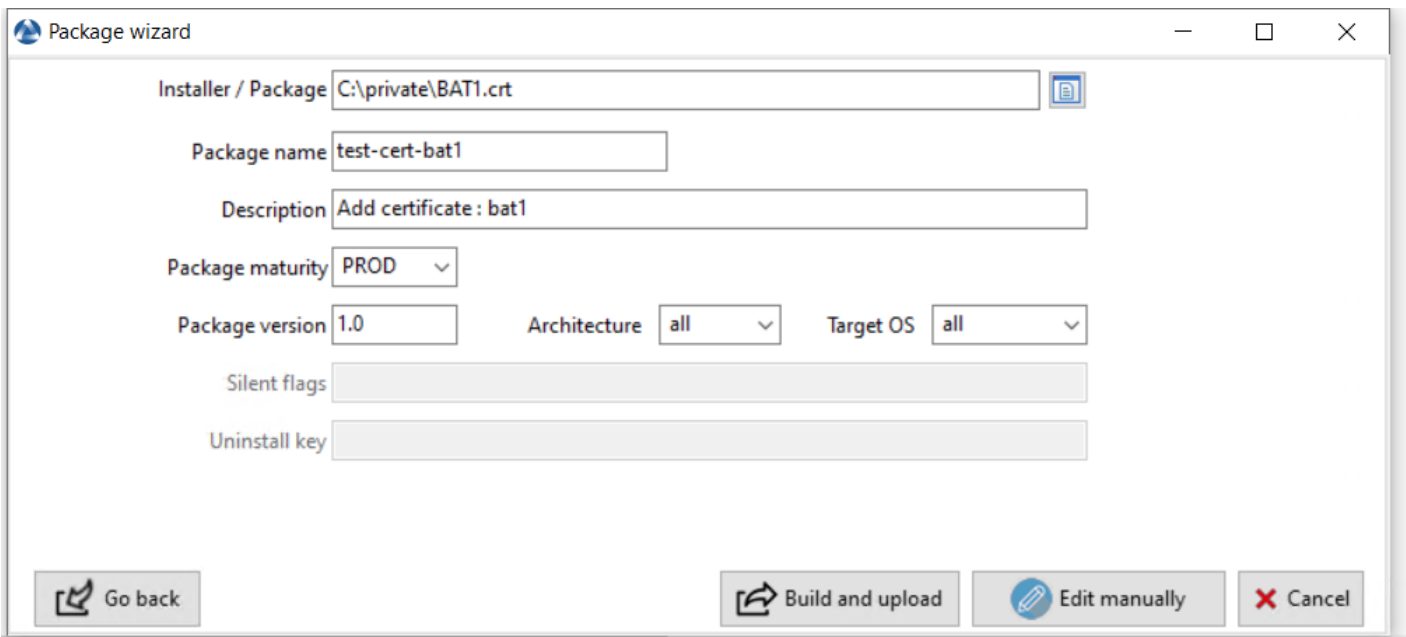
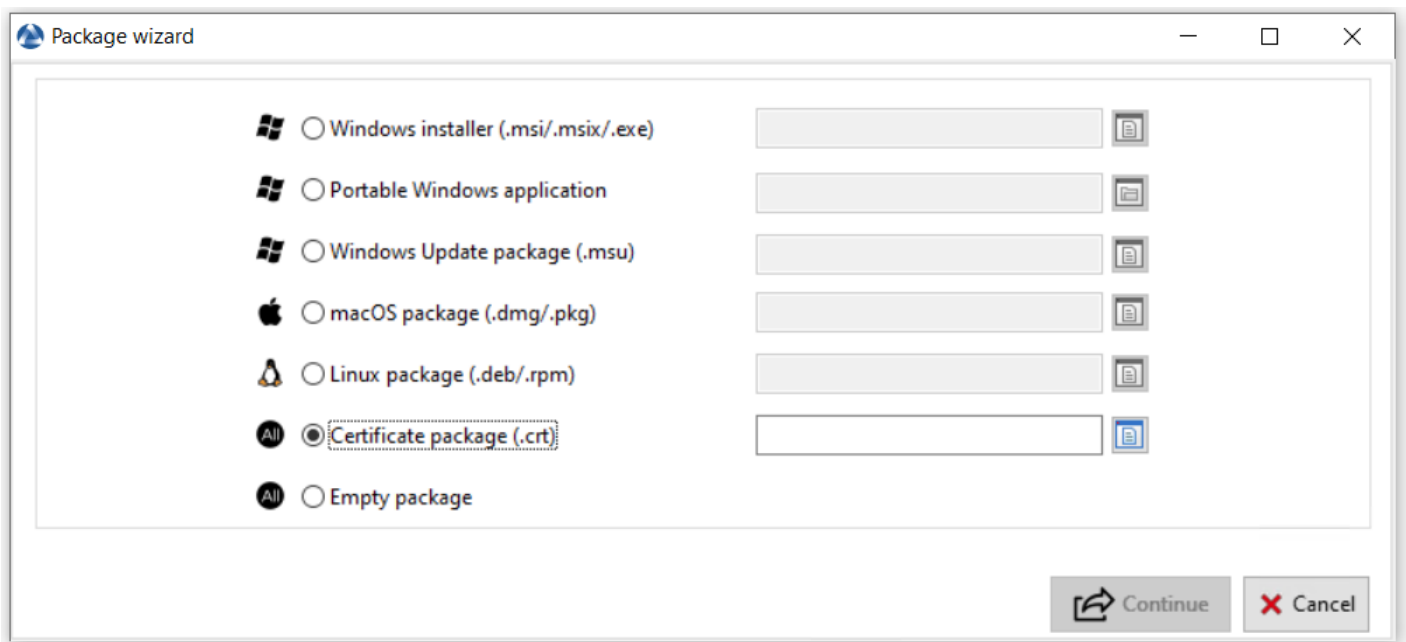
**Paquet vide**

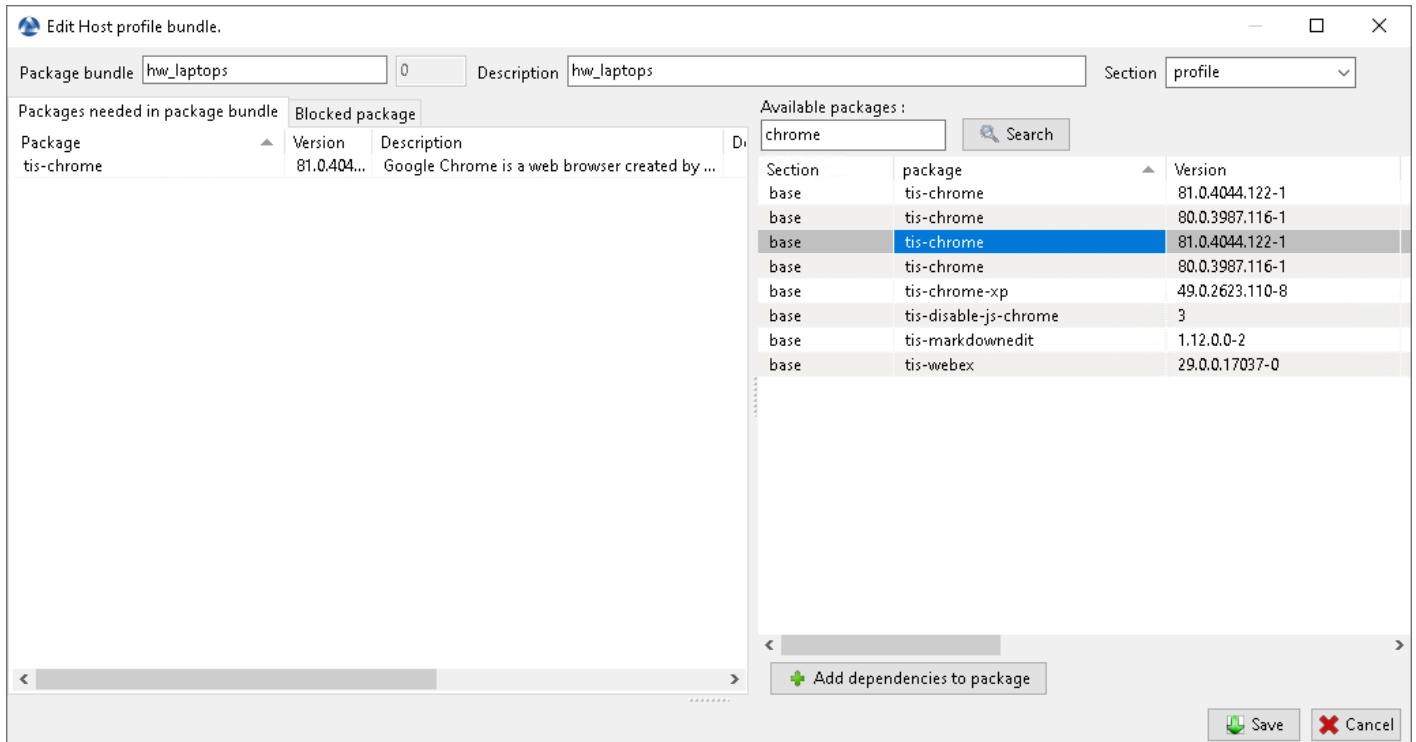
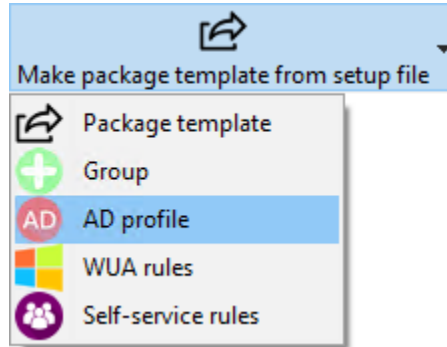
## 65.4.2 Groupe

## 65.4.3 Profile AD

Après avoir cliqué :

Pour son utilisation, voir les *profile AD*





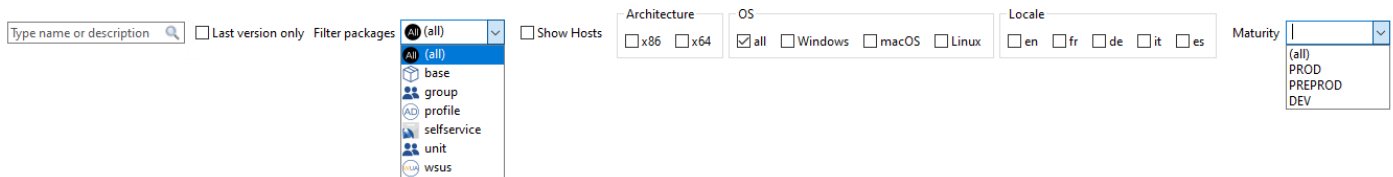
#### 65.4.4 Règles WUA

#### 65.4.5 Règles Self-service

### 65.5 Onglet recherche

### 65.6 Dernière version uniquement

### 65.7 Filtres des paquets



### 65.8 Montrer les hôtes

### 65.9 Architecture

### 65.10 OS

### 65.11 Locale

### 65.12 Maturité

### 65.13 Action sur les paquets

#### 65.13.1 Supprimer du dépôt

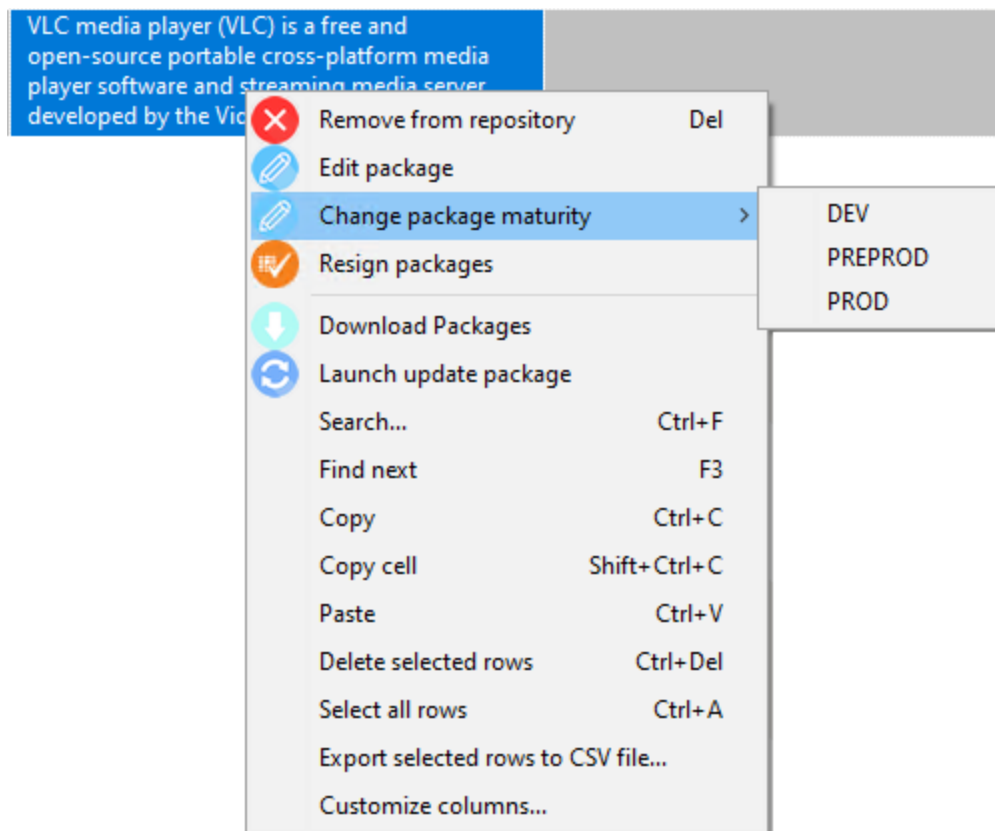
#### 65.13.2 Editer des paquets

#### 65.13.3 Changer la maturité d'un paque

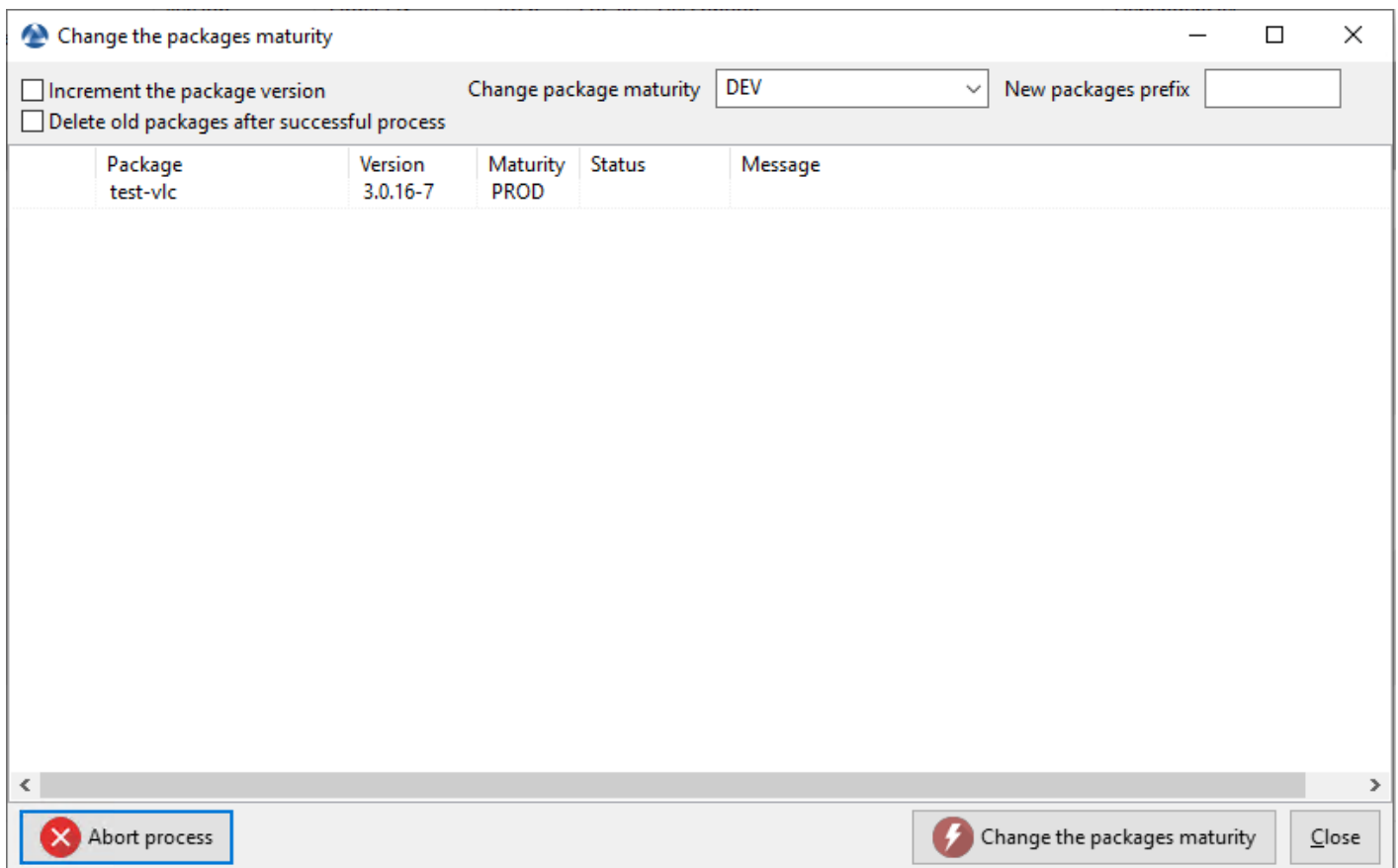
#### 65.13.4 Resigner les paquets

#### 65.13.5 Télécharger les paquets

#### 65.13.6 Rechercher









# CHAPITRE 66

## Onglet Windows Update

WAPT console

File View Tools ?

Inventory WAPT Packages Windows Update Reporting Secondary repos Softwares Inventory

Refresh Download WSUSScan cab from Microsoft Web site Show Download history Delete Unused KB

quilt DevSecOps

Type name or description  Show Hosts Space Used by Windows Updates : 0.0 Mo

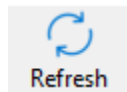
Critical only  
 All products

Published on	KBs	Product	Classification	Title	Severity	Max download size	Downloaded on
--------------	-----	---------	----------------	-------	----------	-------------------	---------------

All classifications

Total : 0 elements

## 66.1 Rafraîchir



Bouton pour rafraîchir l'affichage des mises à jour Windows

## 66.2 Télécharger le cab WSUSScan depuis le site de Microsoft



Bouton pour forcer le téléchargement du cab WSUSScan

## 66.3 Voir l'historique de téléchargement



Bouton pour afficher l'historique de téléchargement

## 66.4 Effacer les mises à jours non-affectées

Bouton pour supprimer les mises à jour Windows inutilisées





## CHAPITRE 67

---

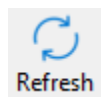
Onglet Source

---





### 68.1 Rafraîchir



Bouton pour rafraîchir l'affichage du reporting

### 68.2 Mode Design



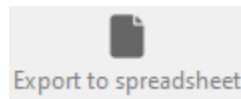
Bouton pour passer en mode conception dans l'onglet reporting

### 68.3 Exécuter



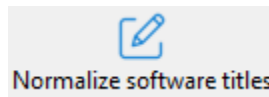
Bouton pour exécuter la requête

## 68.4 Exporter vers tableur



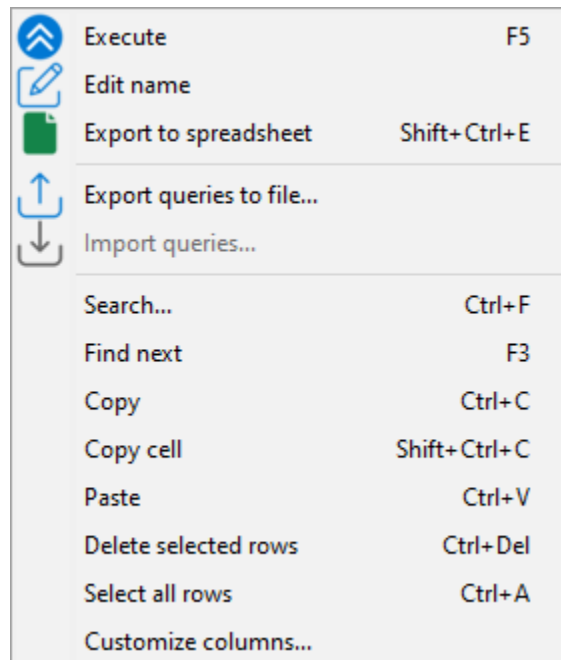
Bouton pour exporter vers un tableur le resultat de la requête

## 68.5 Normaliser les noms de logiciels



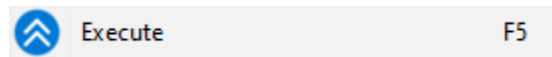
Bouton pour normaliser les noms de logiciels

## 68.6 Menu Clic-droit



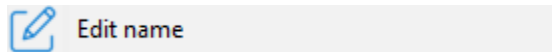
Menu clic-droit sur le panneau de gauche dans l'onglet Reporting

### 68.6.1 Exécuter



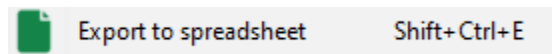
Autre méthode pour exécuter un requête

### 68.6.2 Editer le nom



Renommer le nom de la requête

### 68.6.3 Exporter vers tableur



Autre méthode pour exporter vers un tableur le résultat de la requête

### 68.6.4 Exporter les requêtes vers un fichier


Exporter le résultat des requêtes vers un fichier


### 68.6.5 Importer des requêtes

Importer des requêtes depuis une source externe

### 68.6.6 Rechercher

Rechercher par nom de requête

 Export queries to file...

 Import queries...

Search... Ctrl+F

### 69.1 Onglet de gauche

#### 69.1.1 Rafraîchir



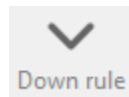
Bouton pour rafraîchir l'affichage des dépôts secondaires

#### 69.1.2 Monter la règle



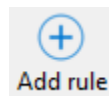
Bouton pour monter la règle dans l'onglet dépôt secondaire

### 69.1.3 Descendre la règle



Bouton pour descendre la règle dans l'onglet dépôt secondaire

### 69.1.4 Ajouter une règle



Bouton pour ajouter une règle dans l'onglet dépôt secondaire

### 69.1.5 Editer la règle



Bouton pour éditer une règle dans l'onglet dépôt secondaire

### 69.1.6 Supprimer la règle

Bouton pour supprimer une règle dans l'onglet dépôt secondaire

### 69.1.7 Sauvegarder les changements

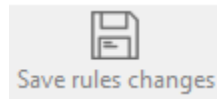
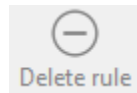
Bouton pour sauvegarder les règles dans l'onglet dépôt secondaire

### 69.1.8 Importer des règles

Bouton pour importer une règle dans l'onglet dépôt secondaire

### 69.1.9 Exporter des règles

Bouton pour exporter une règle dans l'onglet dépôt secondaire



### 69.1.10 Menu Clic-droit

Autre méthode pour éditer une règle sélectionnée

#### Editer la règle

Autre méthode pour éditer une règle sélectionnée

#### Supprimer la règle

Autre méthode pour supprimer la règle sélectionnée

## 69.2 Onglet de droite

### 69.2.1 Rafraîchir

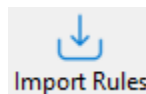
Bouton pour rafraîchir l'affichage des clients des dépôts secondaires

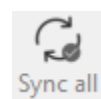
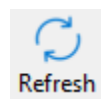
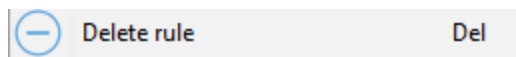
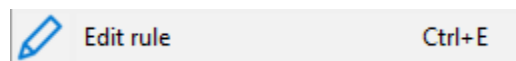
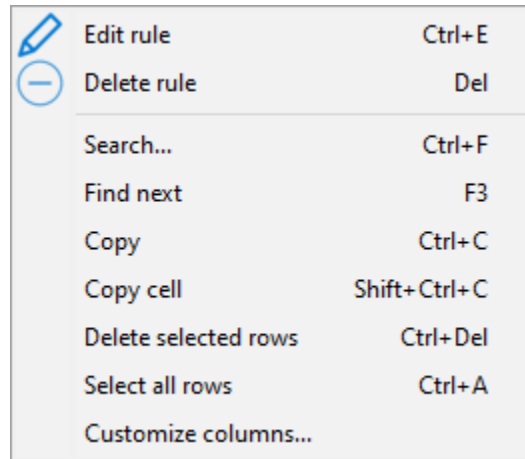
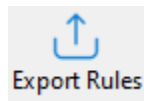
### 69.2.2 Sync sélectionnés

Bouton pour forcer la synchronisation des clients dans l'onglet dépôt secondaire

### 69.2.3 Sync tous

Bouton pour forcer la synchronisation de tous les clients dans l'onglet dépôt secondaire







### 69.2.4 Mettre à jour l'index



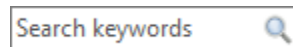
Bouton pour mettre à jour l'index dans l'onglet des dépôts secondaires

### 69.2.5 Changelog



Bouton pour afficher le changelog de l'onglet dépôts secondaires

### 69.2.6 Mots clés de recherche



Rechercher dans les dépôts secondaires des dépôts à l'aide de mots-clés

### 69.2.7 Menu Clic-droit

Menu du clic-droit dans le panneau de droite dans les dépôts secondaires

#### Sync sélectionnés

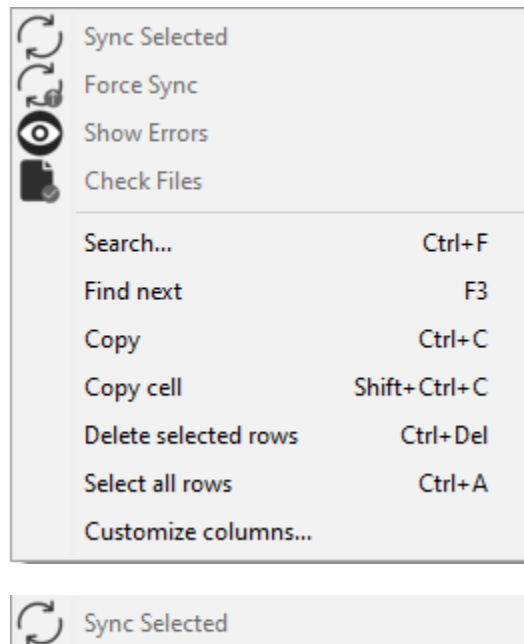
Synchroniser l'hôte sélectionné avec clic-droit dans les dépôts secondaires

#### Forcer la synchronisation

Forcer la synchronisation de l'hôte sélectionné avec clic-droit dans les dépôts secondaires

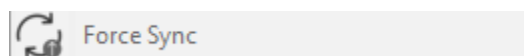
#### Montrer les erreurs

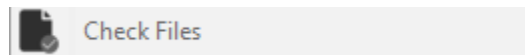
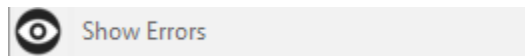
Clic-droit Montrer les erreurs dans les dépôts secondaires sur l'hôte sélectionné



### Vérifier les fichiers

Clic-droit vérifier les fichiers dans les dépôts secondaires sur l'hôte sélectionné







---

## Simplifier le déploiement de vos postes de travail

---

De nombreuses entreprises et administrations incluent des logiciels et des configurations dans les images d'OS qu'elles déploient sur leurs flottes de machines.

Mais désormais ce n'est plus la méthode recommandée pour plusieurs raisons :

- Chaque fois que vous créez une nouvelle image, vous perdez beaucoup de temps à installer un logiciel et à le configurer. Vous êtes très limité dans les paramètres que vous pourrez inclure dans votre image.
- Chaque fois que vous créez une nouvelle image, vous devrez suivre les modifications dans un document texte, une feuille de calcul ou un outil de gestion des modifications.
- Les éditeurs de systèmes d'exploitation (notamment Microsoft) conseillent l'utilisation d'images ISO brutes et leur paramétrage en post-installation.
- Enfin, si vous introduisez dans votre image des configurations de sécurité, des configurations réseau ou des configurations pour limiter l'intrusion de la télémétrie, ces configurations peuvent perturber le fonctionnement normal de WAPT, cela compliquera les diagnostics futurs.

**Avec WAPT, ce n'est plus nécessaire**

### 70.1 Recommandations

Tranquil IT recommande :

- De réaliser une seule image brute par type d'OS avec [MDT](#), [Fog](#) (win10, win2016, etc) ou WAPT WADS sans aucune configuration ou logiciel. **Mettez uniquement les pilotes système** dont vous avez besoin pour le déploiement de votre image dans les répertoires MDT ou Fog prévus à cet effet ;
- Pour créer autant d'Unités Organisationnelles que vous avez de types de machines dans l'OU *CN=Computers* (ex : *standard\_laptop*, *hardened\_laptop*, *workstations*, *servers*, etc) dans votre Active Directory ;
- Pour configurer votre Active Directory afin de distribuer la GPO de l'agent WAPT aux différentes Unités Organisationnelles de machines ; De cette façon, vous pouvez opter pour des configurations fines de votre `waptagent.ini` pour les hôtes rattachés à chaque OU.

---

**Indication :** Pour vous faire gagner du temps, vous pouvez baser votre stratégie de configuration de la sécurité sur les paquets WAPT de sécurité déjà disponibles dans le [WAPT Store](#), vous n'aurez qu'à les compléter en fonction des exigences de sécurité spécifiques

---

de votre Organisation.

---

- Créer dans l'OU *CN=Ordinateurs* autant d'Unités Organisationnelles qu'il y a de types d'utilisation des ordinateurs dans votre organisation (*comptabilité, point\_de\_vente, ingénierie, vente\_sédentaire*, etc).
- Pour créer des paquets WAPT génériques de vos applications logicielles avec leurs configurations associées.

### 70.1.1 Scénario de déploiement

- Vous recevez ou le responsable informatique du site distant reçoit une nouvelle machine dans sa boîte.
- You configure the machine's MAC address in DHCP so that it gets the right system image and is positioned in the right Organizational Unit at the end of the deployment process.
- L'image système attendue est téléchargée sur la machine en temps masqué, la machine est placée dans la bonne Unité Organisationnelle.
- L'agent WAPT enregistre la machine auprès du serveur WAPT, elle apparaît dans la console WAPT.
- L'agent WAPT détecte qu'il se trouve dans une unité organisationnelle qui nécessite un ensemble de logiciels particulier et une configuration de sécurité particulière.
- L'agent WAPT télécharge et exécute des logiciels et des progiciels de configuration de sécurité en temps masqué; l'agent WAPT supprime automatiquement les droits délégués qui sont rendus inutiles après avoir rejoint le domaine pour éviter qu'ils ne soient ensuite exploités de manière non autorisée.
- Soit par groupe de machines ou machine par machine, vous finalisez la configuration des machines en leur attribuant des paquets WAPT spécifiques.

---

**Indication :** Si vous le souhaitez, vous pouvez même laisser l'étape finale de configuration à vos utilisateurs en configurant le libre-service WAPT pour eux (configuration des imprimantes, besoins logiciels spéciaux, etc).

---

### 71.1 Réinitialiser le mot de passe du serveur WAPT Linux

Il arrive parfois de configurer un serveur WAPT et d'oublier son mot de passe.

Pour réinitialiser le mot de passe *SuperAdmin* de la console WAPT, vous devez relancer le processus de post-configuration sur le serveur WAPT.

- se connecter au serveur avec SSH;
- se connecter avec l'utilisateur root (ou utiliser sudo);
- lancer le script de post-configuration..

### 71.2 J'ai perdu ma clé privée WAPT

La sécurité et le bon fonctionnement de WAPT s'appuient sur les jeux de clés privés et de certificats publics.

La perte d'une clé privée nécessite donc de *régénérer une nouvelle clé* et les certificats associés, et ensuite déployer sur le parc de machines les certificats pour la nouvelle clé.

Donc la procédure n'est pas anodine, même si elle est simple.

## 71.2.1 Procédure de renouvellement ou de création d'une clé privée

La procédure va être la suivante :

- *Générer une nouvelle clé privée/certificat public.* Vous conserverez alors la clé privée (fichier *.pem*) dans un endroit sûr ;
- Déployer manuellement avec une GPO ou en utilisant un rôle Ansible (non documenté), le nouveau certificat *.crt* sur vos clients dans le dossier **ssl**.
  - C:\Program Files (x86)\ssl sur Windows ;
  - /opt/wapt/ssl sur Linux et MacOS.

## 71.2.2 Manipulation sur les paquets en conséquence

Les paquets WAPT du dépôt local étant signés avec l'ancienne clé, il convient de re-signer l'intégralité des paquets avec la nouvelle clé.

- *Utiliser la console WAPT*, ou
- *Utiliser la ligne de commande.*

## 71.3 Je me suis fait voler ma clé privée

**Attention : Toute la sécurité de WAPT repose sur la séquestration de cette clé privée.**

WAPT ne gère pas encore la révocation des clés en utilisant une CRL.

La solution consiste à supprimer chaque certificat *.crt* associé à la clé privée volée, situé dans le dossier **ssl** :

- C:\Program Files (x86)\ssl sur Windows ;
- /opt/wapt/ssl sur Linux et MacOS.

Cette opération peut être effectuée à l'aide d'une GPO, manuellement, avec un paquet WAPT ou avec un rôle Ansible (non documenté).

## 71.4 Mon UUID BIOS bogue

- Il arrive parfois qu'un problème survienne avec certains BIOS. WAPT utilise l'*UUID* de la machine comme identifiant pour reconnaître les machines.
- L'*UUID* BIOS est censé être unique, malheureusement chez certains constructeurs et pour certaines séries de machines, les *UUID* des BIOS sont identiques.
- Le PC remontera bien dans la console mais il écrasera le PC déjà présent considérant que l'ordinateur a changé de nom.

### 71.4.1 Résoudre des problèmes de UUID BIOS

WAPT permet de générer un *UUID* aléatoire pour remplacer celui indiqué dans le BIOS.

```
wapt-get generate-uuid
```

Il est aussi possible de ne pas utiliser les *UUID* mais les *FQDN*. Dans le `wapt-get.ini` définissez dans la section `[global]` :

```
use_fqdn_as_uuid = True
```



## 71.5 Mon WAPTdeploy ne fonctionne pas

### 71.5.1 Symptôme

L'utilitaire **waptdeploy** n'arrive pas à installer l'agent WAPT.

### 71.5.2 Lancer WAPTdeploy localement

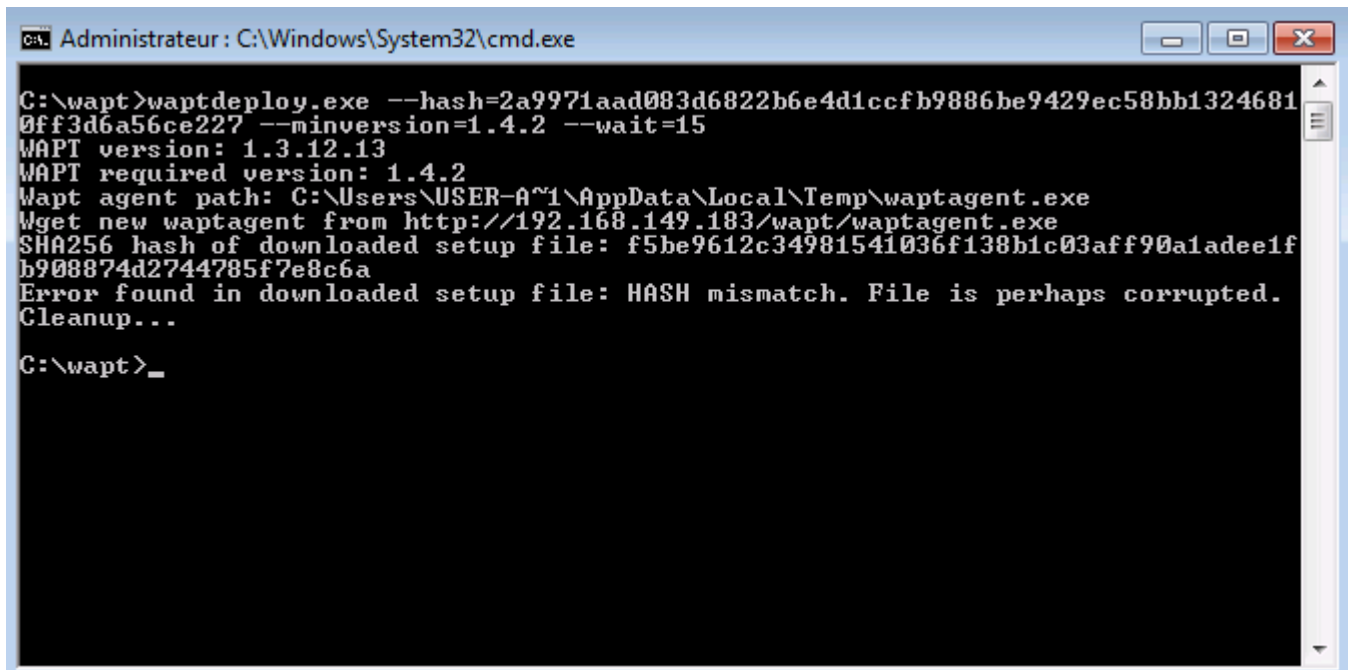
Lancer **waptdeploy** localement peut mettre en évidence un problème en affichant explicitement les erreurs.

**Attention :** Ne pas oublier de lancer l'invite de commande en tant qu'*Administrateur*.

Exemple de commande à lancer :

```
C:\Program Files (x86)\wapt\waptdeploy.exe --  
↪hash=2a9971aad083d6822b6e4d1ccfb9886be9429ec58bb13246810ff3d6a56ce887 --minversion=2.1.0.10550 --  
↪wait=15 --waptsetupurl=https://srvwapt.mydomain.lan/wapt/waptagent.exe
```

Dans notre cas le hash n'est pas le bon.



```
Administrator: C:\Windows\System32\cmd.exe  
C:\wapt>waptdeploy.exe --hash=2a9971aad083d6822b6e4d1ccfb9886be9429ec58bb1324681  
0ff3d6a56ce227 --minversion=1.4.2 --wait=15  
WAPT version: 1.3.12.13  
WAPT required version: 1.4.2  
Wapt agent path: C:\Users\USER-A~1\AppData\Local\Temp\waptagent.exe  
Wget new waptagent from http://192.168.149.183/wapt/waptagent.exe  
SHA256 hash of downloaded setup file: f5be9612c34981541036f138b1c03aff90a1adee1f  
b908874d2744785f7e8c6a  
Error found in downloaded setup file: HASH mismatch. File is perhaps corrupted.  
Cleanup...  
C:\wapt>_
```

FIG. 1 – Error with WAPTDeploy

## WAPTdeploy fonctionne manuellement mais ne fonctionne pas via GPO

Vérifier que la GPO est bien appliquée avec la commande :

```
gpresult /h gpo.html & gpo.html
```

Pour forcer l'application des GPO :

```
gpupdate /force
```

Si **waptdeploy** n'apparaît pas, il faut re-vérifier la configuration de la GPO :

- Il est possible que vous utilisiez une ancienne version du `program : waptdeploy`, :download :téléchargez la dernière version <<https://store.wapt.fr/wapt/releases/latest/waptdeploy.exe>> de **waptdeploy** sur la page WEB de votre serveur WAPT.
- Merci à Emmanuel EUGENE de l'institution française **INSERM** qui a soumis cette cause possible du mauvais fonctionnement du **waptdeploy**, si vous répliquez des contrôleurs de domaine, assurez-vous que les GPOs sont correctement synchronisées entre vos DCs et que les ACLs sont appliquées de manière identique sur le SysVols.

## 71.6 Windows n'attend pas le réseau au démarrage

Par défaut Windows n'attend pas le réseau au démarrage de la machine.

Cela peut poser soucis pour l'exécution de **waptdeploy** car celui-ci a besoin du réseau au démarrage pour télécharger l'agent WAPT.

2 solutions :

1. Nous vous recommandons d'ajouter `waptdeploy.exe` aux script de démarrage et d'extinction *sur la GPO*.
2. Vous pouvez activer la GPO : **Toujours attendre le réseau au démarrage et à la connexion** :

Computer → Configuration → Administrative Templates → System → Logon → Always wait for the network at computer startup and logon

The screenshot shows the 'Local Computer Policy' console with the 'Logon' policy 'Always wait for the network at computer startup and logon' selected. The policy is currently set to 'Enabled'. A table below lists other logon policies and their states.

Setting	State	Comment
Assign a default domain for logon	Not configured	No
Exclude credential providers	Not configured	No
Do not process the legacy run list	Not configured	No
Do not process the run once list	Not configured	No
Turn off Windows Startup Sound	Not configured	No
Hide entry points for Fast User Switching	Not configured	No
Always use classic logon	Not configured	No
Don't display the Getting Started welcome screen at logon	Not configured	No
Run these programs at user logon	Not configured	No
<b>Always wait for the network at computer startup and logon</b>	<b>Enabled</b>	No
Always use custom logon background	Not configured	No

## 71.7 WAPTextit ne se lance pas

Malgré la présence du script dans les stratégies locales d'arrêt de la machine, **waptexit.exe** ne se lance pas à l'extinction du poste.

### 71.7.1 Résolution : Hybrid Shutdown

Il faut désactiver l'arrêt hybride de Windows10, qui cause par ailleurs beaucoup d'autres comportements étranges. La désactivation de l'arrêt hybride rétablit l'exécution des scripts à l'extinction de la machine.

L'arrêt hybride peut être désactivé en précisant une valeur dans le fichier `wapt-get.ini` de l'agent WAPT, voir *paramètres waptexit*.

It's possible to set this value when *creating WAPT Agent*.

There is a WAPT package to solve the Hybrid Shutdown problem : [tis-disable-hybrid-shutdown](#).

### 71.7.2 Windows édition familiale

Les GPO n'étant pas présentes sur les versions familiales de Windows, il est normal qu'elles ne s'exécutent pas.

La solution de contournement est d'utiliser une tâche planifiée qui appelle `C:\Program Files (x86)\wapt\wapt-get.exe` avec le paramètre `upgrade`.

### 71.7.3 Corrupted local GPO

Il peut arriver que les stratégies de groupes locales de la machine soient corrompues.

One of the possible solutions is to :

- Remove local security strategies by deleting the file `C:\Windows\System32\GroupPolicy\gpt.ini` ;
- Restart the computer ;
- Re-install the shutdown scheduled tasks by :

```
wapt-get add-upgrade-shutdown
```

Si le problème se reproduit, cela signifie peut être qu'une autre application manipule également la GPO.

## 71.8 WAPTextit se coupe après 15 minutes et n'achève pas l'installation

Par défaut sous Windows, les scripts d'extinction ne peuvent s'exécuter plus de 15 minutes.

Si à l'arrêt de la machine, un script d'extinction n'a pas rendu la main au bout de 15 minutes, le script est interrompu.

## 71.8.1 Solution : augmenter le délai d'installation

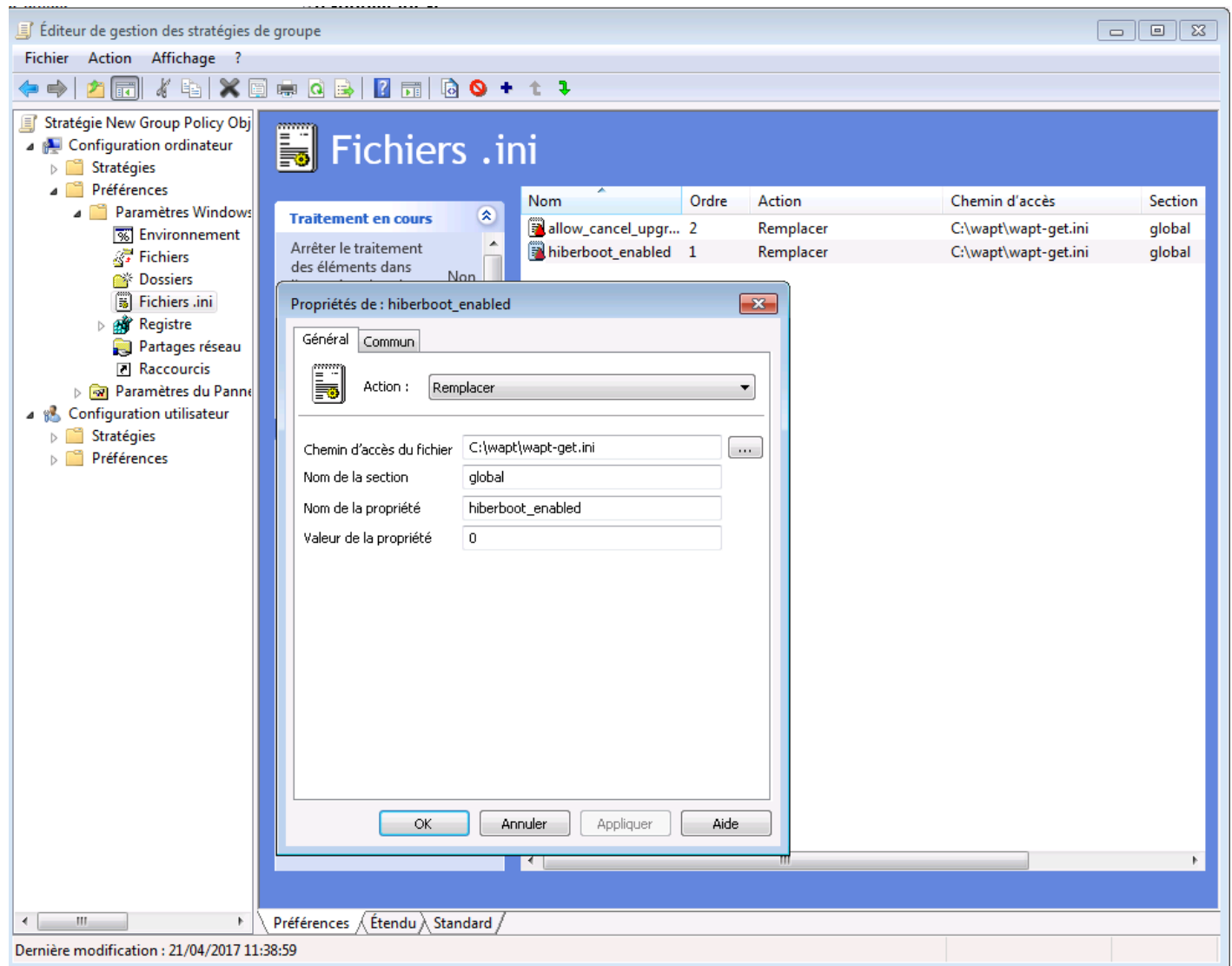
To solve that problem, increase the `pre_shutdown_timeout` `` value and the `max_gpo_script_wait` value on `wapt-get.ini`

Define *these values* to change the default behavior.

```
max_gpo_script_wait=180
pre_shutdown_timeout=180
```

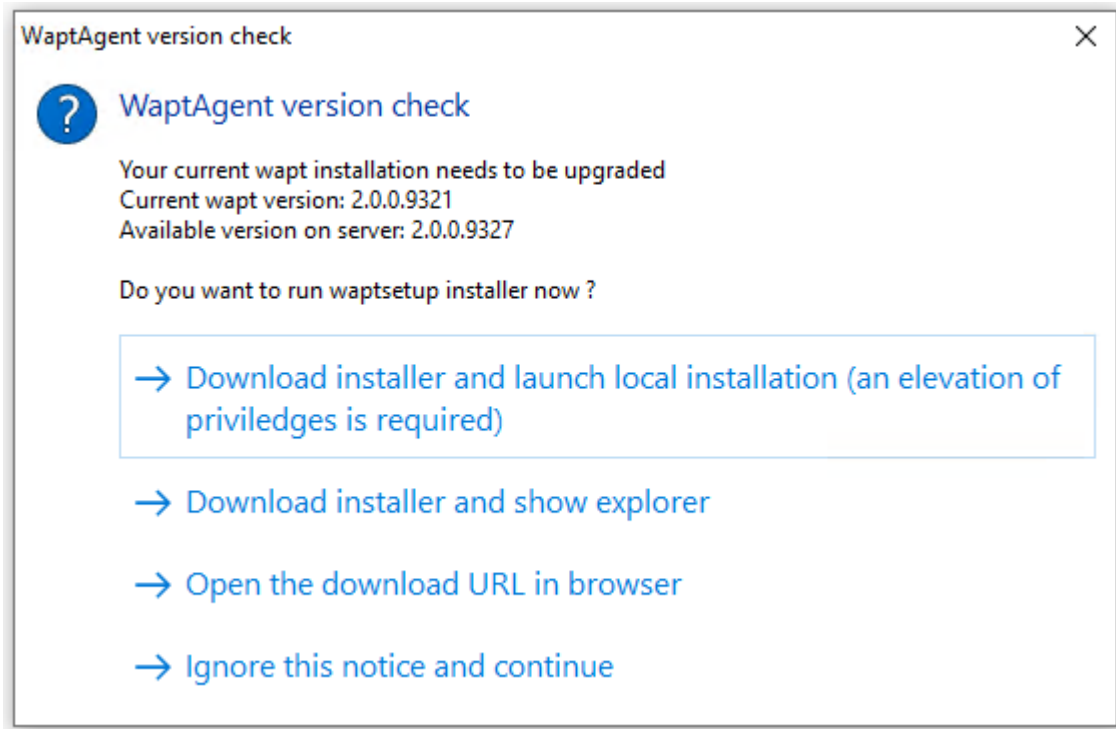
Le paquet `tis-wapt-conf-policy` embarque cette configuration.

L'autre solution est d'utiliser la GPO `File.ini`.



## 71.9 Message d'erreur à l'ouverture de la console

### 71.9.1 Vérification de la version



La version de la console WAPT n'est pas égale à la version du serveur. Il est recommandé de la mettre à jour. Sélectionnez votre méthode préférée.

### 71.9.2 Connection refused

La console WAPT ne parvient pas à joindre le port 443 du serveur.

— Vérifier si le service **Nginx** est démarré sur le serveur.

```
systemctl status nginx
```

— Si **Nginx** n'est pas lancé, relancer **Nginx**.

```
systemctl restart nginx
```

— If **Nginx** still does not start, you'll need to analyze journal logs in :

— /var/log/nginx/ on Linux

— C:\Program Files (x86)\wapt\waptserver\nnginx\logs sur Windows.

### 71.9.3 Service Unavailable

Il est possible que le service *waptserver* soit stoppé.

- Vérifier si le service **waptserver** est en cours d'exécution.

```
systemctl status waptserver
```

- If the command returns an error, then start the **waptserver**.

```
systemctl start waptserver
```

### 71.9.4 Error connecting with SSL ... verify failed

La console ne semble pas réussir à vérifier le certificat HTTPS du serveur.

**Attention :** Attention, avant toute chose vérifiez que vous n'êtes pas victime d'une attaque *man in the middle* !

---

**Note :** Si vous venez de refaire votre serveur WAPT et que vous utilisez un certificat auto-signé, vous pouvez récupérer les anciennes clés de votre ancien serveur wapt dans `/opt/wapt/waptserver/apache/ssl`.

---

- Fermer votre console WAPT.
- Supprimer le dossier `%appdata%\..\Local\waptconsole`.
- Lancer ensuite la commande `wapt-get enable-check-certificate`.
- Assurez-vous que la commande précédente s'est bien déroulée.
- Redémarrer le service WAPT avec `net stop waptservice && net start waptservice`.
- Relancer la console WAPT.

Dans le cas où vous ne pratiquez pas le *certificate pinning*, cela signifie que le certificat envoyé par le serveur ne pas être vérifié avec le bundle python **certifi**. Veillez à bien fournir la chaîne complète pour le certificat sur le serveur WAPT.

## 71.10 Problèmes pour enregistrer une machine avec le serveur WAPT

Si vous faites un `wapt-get register` et que la commande renvoie :

```
FATAL ERROR : ConnectionError: HTTPSConnectionPool(host='XXX.XXX.XXX.XXX', port=443): Max retries_
↪ exceeded with url: /add_host
```

Vous devez vérifier que le port 443 est correctement transmis au serveur WAPT et qu'il n'est pas bloqué par un pare-feu.

## 71.11 Problème lors du enable-check-certificate

### 71.11.1 J'ai le message « certificate CN ### sent by server does not match URL host ### lors du enable-check-certificate »

Cela signifie que le CN envoyé par le certificat du serveur ne correspond pas au *wapt\_server* du fichier *wapt-get.ini*.

— Deux solutions :

1. Vérifier le paramètre *wapt\_server* dans votre fichier *wapt-get.ini*.

Si votre valeur est correcte, cela signifie sûrement qu'une erreur est survenue lors de la génération du certificat autosigné par le post conf, une faute de frappe ...

Vous pouvez donc régénérer vos certificats autosignés.

2. Sur le serveur WAPT, supprimez le contenu du dossier */opt/wapt/waptserver/apache/ssl/*.

Ensuite, relancez le script de postconf (le même que pendant l'installation initiale, avec les mêmes arguments et les mêmes valeurs).

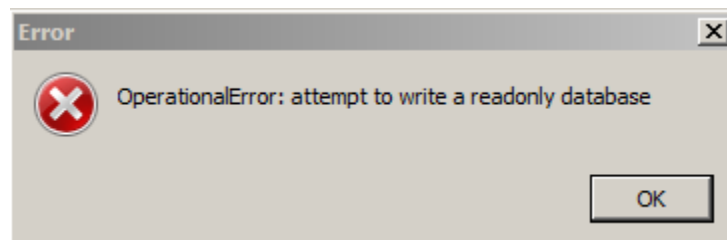
Enfin, vérifiez bien le nom renseigné lors de l'étape *FQDN for the WAPT serveur* est correcte.

— Vous pouvez maintenant retenter votre **enable-check-certificate**.

## 71.12 Problème avec la création de paquet

### 71.12.1 Problème de droits avec PyScripter

Lorsqu'on souhaite tester l'installation d'un paquet sur son PC de développement de paquet à partir de **PyScripter**, on obtient le message :



#### Solution

Ouvrir une session en tant qu'Administrateur Local et refaire l'opération souhaitée.

## 71.12.2 Mon paquet WAPT est trop volumineux et je n'arrive pas à l'uploader

Quand un paquet est trop volumineux, il faut en général lancer le builder localement puis l'uploader avec **WinSCP**.

### Solution

- Assembler le paquet dans PyScripter ou *manuellement*.

---

**Indication :** Si l'**upload** a précédemment échoué, vous pourrez trouver le paquet généré dans C:\waptdev.

---

- Télécharger et installer **WinSCP**
- En utilisant **WinSCP**, **uploadez** votre paquet dans *le dépôt*.
- Une fois le téléchargement terminé, vous devrez recréer le fichier d'index Packages sur votre dépôt. Utilisez la commande suivante pour remplacer **repository** par *l'emplacement du dépôt*.

```
wapt-scanpackages repository
```

## 71.12.3 Erreur de violation d'accès lors de la re-signature du paquet

 microsoft-office 16.0.12325.20276-2 PROD ERROR Access violation

Si l'erreur **Access violation** apparaît, c'est parce que le paquet est trop gros.

Éditez le paquet et suivez *cette procédure*.

## 71.12.4 Paquet WAPT en erreur

### Problème d'installation

### Symptôme

J'ai un paquet en erreur et le logiciel n'est pas installé sur la machine quand je me déplace.

### Explication

Une erreur est survenue pendant l'exécution de l'installation définie dans `setup.py`.

Vous pouvez lire et analyser les messages d'erreur retournés dans la console et tenter de les comprendre.

L'installation sera retentée à chaque **upgrade** jusqu'à ce que le paquet ne génère plus d'erreur.



## Solution

- Si WAPT fournit un code d'erreur, chercher ce code d'erreur sur Internet.  
Exemple with a MSI : *1618* : Une autre installation est déjà en cours. Un redémarrage devrait solutionner le problème.

---

**Note :** Les différents codes d'erreur MSI sont disponibles [ici](#).

---

- Se déplacer physiquement sur la machine en erreur et relancer l'installation silencieuse en ligne de commande. Vérifier ensuite que le logiciel à bien été installé.

**Attention :** Une fois l'installation silencieuse lancée, ne pas intervenir.

L'objectif est de reproduire le comportement de l'agent WAPT.

- Si l'installation fonctionne en mode silencieux, en contexte utilisateur, cela peut signifier que l'installateur ne supporte pas l'installation en compte *SYSTEM*.
- Si cela ne fonctionne toujours pas, lancer l'installation manuellement. Il est possible qu'une erreur apparaisse indiquant explicitement le problème. Exemple (Dépendance manquante : .Net , Java, etc..).
- Il est possible que l'installateur ne supporte pas l'écrasement d'une installation précédente, alors prévoir la désinstallation des anciennes version avant d'installer la nouvelle version .

## Erreur « timed out after seconds with output “600.0” »

### Symptôme

Certains paquets dans la console retournent l'erreur :

```
"Erreur timed out after seconds with output '600.0'"
```

### Explication

Par défaut lors d'une installation (avec `run`, `install_msi_if_needed` ou `install_exe_if_needed`) WAPT va attendre 600 secondes que l'installateur lui rende la main.

si l'installateur n'a pas terminé dans ce délai, WAPT coupera l'installation en cours.

### Solution

Si vous tentez d'installer un gros logiciel (Office, Solidworks, Libreoffice ...), il est possible que l'intervalle de 600 secondes ne soit pas suffisant.

Vous devez alors, augmenter cette valeur avec l'argument `timeout`, ex : `timeout = 1200` :

```
run('"setup.exe" /adminfile office2010noreboot.MSP', timeout=1200)
```

### Erreur « has been installed but the uninstall key can not be found »

#### Symptôme

Certains paquets dans la console retournent l'erreur :

```
XXX has been installed but the uninstall key can not be found.
```

#### Explication

WAPT s'appuie sur Windows pour installer les binaires `.msi` avec `install_msi_if_need` et les binaires `.exe` avec `install_exe_if_need`.

Par défaut, WAPT accepte les codes de retour : `0` (OK) et `3010` (redémarrage nécessaire), et il vérifie la clé de désinstallation résultante (*uninstall key*).

Malheureusement, on ne peut pas toujours se fier à ces codes d'erreur, WAPT vérifie enfin que tout s'est bien déroulé :

- Il vérifie la présence de la clé de désinstallation sur la machine.
- Il vérifie que la version est bien égale ou supérieure à celle renseignée du fichier `control`.
- Si ce n'est pas le cas, il en déduit que le logiciel n'est peut-être pas présent sur la machine.

La fonction bascule alors volontairement le paquet en erreur. L'installation sera retentée lors de chaque upgrade, jusqu'à ce que le paquet ne génère plus d'erreur.

#### Solution

**Attention :** Avant toute chose, il convient de se connecter sur la machine en erreur et de vérifier manuellement **si le logiciel est correctement installé** . Si ce n'est pas le cas, se référer à la documentation sur les *problèmes d'installation d'un paquet*..

- Si le logiciel est bien installé, cela signifie peut être que la clé de désinstallation ou la version fournie dans le paquet n'est pas bonne.
- Récupérer la bonne clé de désinstallation et corriger le paquet en conséquence.
- Si l'erreur se produit avec `install_msi_if_needed` cela signifie que l'installeur MSI est mal conçu et renvoie une mauvaise *clé de désinstallation*.

### Erreur « has been installed and the uninstall key found but version is not good »

#### Symptôme

Certains paquets dans la console retournent l'erreur :

```
has been installed and the *uninstall key* found but version is not good
```

## Explication

Avec les commandes `install_msi_if_needed` et `install_exe_if_needed`, des vérifications supplémentaires sont effectuées pour vérifier que tout s'est bien passé.

## Solution

**Attention :** Avant toute chose, il convient de se connecter sur la machine en erreur et de vérifier manuellement **si le logiciel est correctement installé** . Si ce n'est pas le cas, se référer à la documentation sur les *problèmes d'installation d'un paquet*..

### Solution : Avec `install_msi_if_needed`

Les informations étant extraites depuis l'installeur MSI, cela signifie que le fichiers MSI ne renvoie pas la bonne version ou que la clé de désinstallation retournée n'est pas la bonne version.

Vérifier avec la commande :

```
wapt-get list-registry
```

Si la clé retournée n'est pas celle renseignée dans la partie installation du fichier `setup.py` , il n'est pas possible d'utiliser la fonction `install_msi_if_needed`.

Il faut rebasculer l'installation avec un simple `run()` et gérer les exceptions manuellement.

### Avec `install_exe_if_needed`

Cela signifie probablement que la version renseignée dans la fonction `install_exe_if_needed` n'est pas la bonne. Corriger le paquet WAPT en conséquence.

---

**Note :** Si l'argument `min_version` n'a pas été renseigné, WAPT va tenter de récupérer automatiquement la version depuis l'installeur `exe`.

---

Pour vérifier la clé de désinstallation utilisée et le numéro de version, utiliser la commande :

```
wapt-get list-registry
```

Si aucune version n'est fournie avec la commande `list-registry`, cela signifie que la clé de désinstallation du logiciel ne fournit pas de version.

Deux solutions :

- Utiliser l'argument `get_version` pour fournir un chemin vers une autre `uninstallkey`.

```
def install():
    def versnaps2(key):
        return key['name'].replace('NAPS2 ', '')
```

(suite sur la page suivante)

(suite de la page précédente)

```
install_exe_if_needed('naps2-5.3.3-setup.exe', silentflags='/VERYSILENT', key='NAPS2 (Not Another PDF Scanner 2)_is1', get_version=versnaps2)
```

— Fournir une valeur vide en argument pour `min_version` afin d'indiquer à WAPT qu'aucune version n'est à vérifier.

```
min_version= ' '
```

**Attention :** Avec cette méthode **on ne vérifiera plus la version lors de mise à jour !**

## 71.13 Problèmes fréquents liés aux Antivirus

Certains Antivirus lèvent des alertes pour des composants de WAPT.

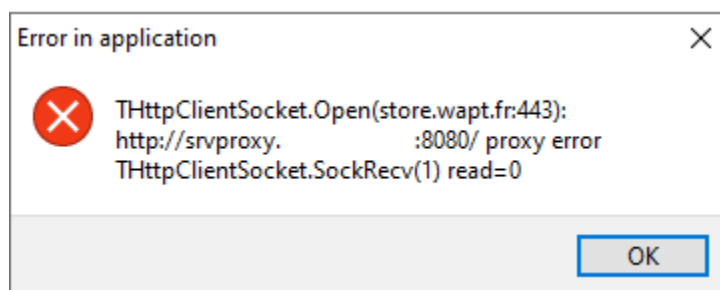
Parmi ceux-ci le composant **nssm.exe** est utilisé par WAPT comme utilitaire de service pour l'agent WAPT.

Voici une liste des exceptions possibles à déclarer dans votre interface de gestion centralisé antivirus :

```
"C:\Program Files (x86)\wapt\waptservice\win32\nssm.exe"
"C:\Program Files (x86)\wapt\waptservice\win64\nssm.exe"
"C:\Program Files (x86)\wapt\waptagent.exe"
"C:\Program Files (x86)\wapt\waptconsole.exe"
"C:\Program Files (x86)\wapt\waptexit.exe"
"C:\wapt\waptservice\win32\nssm.exe"
"C:\wapt\waptservice\win64\nssm.exe"
"C:\wapt\waptagent.exe"
"C:\wapt\waptconsole.exe"
"C:\wapt\waptexit.exe"
"C:\Windows\Temp\waptdeploy.exe"
"C:\Windows\Temp\waptagent.exe"
"C:\Windows\Temp\is-?????.tmp\waptagent.tmp"
```

## 71.14 Je rencontre un problème avec mon proxy - THttpClientSocket.SockRecv(1) read = 0

Si vous avez ce problème :



L'erreur vient d'une option *timeout* dans votre `waptconsole.ini`.

En effet, depuis la version WAPT 2.1, le `timeout` est défini en millisecondes et non pas en secondes comme avant.

Vous allez devoir supprimer votre option *timeout* dans votre Console WAPT situé ici : `%localappdata%\waptconsole`.



### 72.1 Comment déplacer mon dépôt sur une autre partition

Pour de multiples raisons, vous devrez éventuellement déplacer le référentiel sur une autre partition.

Votre dépôt contient 3 dossiers qui peuvent être assez volumineux :

- `wapt` ;
- `wapt-host` ;
- `waptwua` ;

**Avertissement :** Nous n'utiliserons pas la même méthode pour Linux et Windows.

#### 72.1.1 Linux

Sous Linux, créez un point de montage sur `fstab`.

Dans cet exemple, la deuxième partition est nommée `part2`.

`part2` est une partition **formatée en ext4**.

### Debian / Ubuntu

- Créer le dossier temporaire.

```
mkdir /mnt/tmp
```

- Création d'un point de montage temporaire.

```
mount /dev/part2 /mnt/tmp
```

- Déplacer les dossiers.

```
mv /var/www /mnt/tmp
```

- Démonter la partition.

```
umount /dev/part2
```

- Modifier le fichier `fstab`.

```
vi /etc/fstab
```

- Ajouter la ligne suivante au fichier `fstab`.

```
# <file system> <mount point> <type> <options> <dump> <pass>  
/dev/part2 /var/www ext4 defaults 0 0
```

- Monter la partition.

```
mount -a
```

---

**Indication :** Si aucune erreur, la partition est montée.

---

- Vous pouvez vérifier en exécutant.

```
df -h  
  
#Result  
Filesystem      1K-blocks      Used Available Use% Mounted on  
dev/part2       15G            944M        14G    7% /var/www
```

- Supprimer le dossier temporaire.

```
rm -rf mnt/tmp
```



**Centos / RedHat**

- Créer un dossier temporaire pour copier les dossiers.

```
mkdir /mnt/tmp
```

- Création d'un point de montage temporaire.

```
mount /dev/part2 /mnt/tmp
```

- Déplacer les dossiers.

```
mv /var/www/html /mnt/tmp
```

- Démonter la partition.

```
umount /dev/part2
```

- Modifier le fichier `fstab`.

```
vi /etc/fstab
```

- Ajouter la ligne suivante au fichier `fstab`.

```
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/part2      /var/www/html      ext4      defaults 0      0
```

- Monter la partition.

```
mount -a
```

---

**Indication :** Si aucune erreur, la partition est montée.

---

- Vous pouvez vérifier en exécutant.

```
df -h

#Result
Filesystem      1K-blocks      Used Available Use% Mounted on
dev/part2       15G             944M      14G    7% /var/www
```

- Supprimer le dossier temporaire.

```
rm -rf mnt/tmp
```

## Windows

Sous Windows, la meilleure méthode est de *sauvegarder* et *restaurer* le serveur sur la nouvelle partition.

---

**Note :** Il est possible d'installer le serveur sur une autre partition que C:.

---

## 72.2 Utiliser un lecteur réseau pour stocker et livrer des paquets WAPT

Le mode de fonctionnement standard de WAPT est avec un serveur Web sécurisé qui fournit les paquets WAPT aux clients WAPT.

**Tranquil IT déconseille l'utilisation d'un lecteur réseau pour la livraison de paquets WAPT** pour plusieurs raisons :

- Un serveur web est extrêmement facile à installer, à sécuriser, à maintenir, à sauvegarder et à surveiller.
- Pour fonctionner correctement, un paquet WAPT doit être autonome. En effet, nous ne savons pas si le réseau sera disponible au moment du lancement de l'installation (par exemple si nous avons un waptexit qui démarre lorsque la station de travail s'arrête sur un réseau avec une authentification utilisateur 802.1x, il n'y aura plus de réseau disponible au moment de l'installation). La nature autonome du WAPT le rend plus déterministe que les autres solutions de déploiement.
- Une congestion du réseau peut résulter du téléchargement de gros paquets sur de grandes flottes d'appareils parce que vous avez moins de contrôle sur la consommation de bande passante ou bien vous ne pouvez pas terminer un téléchargement partiel.
- Cette méthode casse ou au moins affaiblit le cadre de sécurité du WAPT.
- Cette méthode ne vous permet pas d'exposer vos dépôts sur Internet pour votre personnel itinérant.

**Attention :** Même si WAPT *peut fonctionner* indépendamment du mode de transport, **Tranquil IT ne supportera pas officiellement l'utilisation d'un lecteur réseau pour stocker et livrer des paquets WAPT.**

## 72.3 Utiliser la fonction register() dans vos scripts d'audit

La fonction register() force l'envoi au serveur WAPT de l'inventaire matériel et logiciel de l'agent WAPT.

Cette fonction est très éprouvante pour les performances du serveur car elle oblige le serveur à analyser un JSON (Java Script Object Notation) BLOB (Binary Large Object) relativement grand et à injecter le résultat dans la base de données PostgreSQL.

La fonction est par défaut déclenchée manuellement ou lorsqu'une nouvelle mise à niveau de paquet est appliquée.

Lorsque vous utilisez la fonction register() dans un script d'audit, elle sera exécutée à chaque fois que le script d'audit est déclenché et chargera le serveur sans bénéfice apparent.

Par conséquent, **nous ne recommandons pas l'utilisation de la fonction register() dans les scripts d'audit.**

## 72.4 EWaptBadControl : "utf8" codec can't decode byte

Si vous recevez ce message, cela peut signifier que vous n'avez pas mis en place correctement votre environnement de développement. Visitez cette section de la documentation sur la configuration de l'UTF-8 (pas de BOM).

## 72.5 J'ai bien plus d'hôtes dans la console que de paquets host sur mon serveur ?

Suite à une remarque de Philippe LEMAIRE du [Lycée Français Alexandre Yersin](#) à Hanoï, si vous utilisez la version Entreprise du WAPT et que vous faites un usage intensif des paquets *unit* ou *profile packages*, vous pouvez réaliser que vous aurez beaucoup plus d'hôtes dans votre console que de *\*host packages\** sur votre serveur WAPT. **C'est normal.**

En fait, les paquets *unit* et les paquets *profile* ne sont pas explicitement attribués à la machine (c'est-à-dire comme des dépendances dans le paquet *host*) mais sont implicitement pris en compte par le moteur de dépendance de l'agent WAPT lors de la mise à niveau WAPT.

On peut donc ne pas avoir de paquet *host* sur le serveur si seuls des paquets *unit* sont utilisés pour gérer une flotte d'appareils.



---

## Installer le serveur WAPT avec Ansible

---

Pour éviter les erreurs et automatiser le déploiement de votre serveur WAPT, nous fournissons des rôles Ansible pour l'installation du serveur WAPT.

Vous pouvez explorer le code source du rôle en [visitant le dépôt Tranquil IT sur Github](#).

### 73.1 Pré-requis

- hôtes Debian Linux ou CentOS ;
- un sudoer account sur ces machines ;
- Ansible 2.8.

### 73.2 Installer le rôle Ansible

#### 73.2.1 Discovery

#### 73.2.2 Enterprise

- Installer le rôle Ansible `tranquilit.waptagent`.

```
ansible-galaxy install tranquilit.waptserver
```

- Pour installer le rôle ailleurs, utilisez la sous-commande `-p` comme ceci.

```
ansible-galaxy install tranquilit.waptserver -p /path/to/role/directory/
```

## 73.3 Utiliser le rôle Ansible

- Assurez-vous d'avoir une clé ssh fonctionnelle déployée sur vos hôtes, sinon vous pouvez en générer et la copier comme indiqué ci-dessous.

```
ssh-keygen -t ed25519
ssh-copy-id -i id_ed25519.pub user@srvwapt.mydomain.lan
ssh user@srvwapt.mydomain.lan -i id_ed25519.pub
```

- Editer l'inventaire ANsible des machines ( `./hosts` ) et ajouter les machines Linux.

```
[srvwapt]
srvwapt.mydomain.lan ansible_host=192.168.1.40
```

- Créer un *playbook* avec le contenu suivant dans `./playbooks/deploywaptagent.yml`.

```
- hosts: srvwapt
  roles:
    - { role: tranquilit.waptserver }
```

- Lancez votre *playbook* avec la commande suivante.

```
ansible-playbook -i ./hosts ./playbooks/wapt.yml -u user --become --become-method=sudo -K
```

Le serveur est maintenant prêt. Vous pouvez aller à la documentation pour *Installer la console WAPT !!*

### 73.3.1 Paramètres des rôles Ansible

Les variables disponibles sont énumérées ci-dessous, ainsi que les valeurs par défaut (voir `defaults/main.yml`) :

- Version de WAPT qui sera installée à partir du dépôt WAPT Deb/RPM.

```
wapt_version: "2.0"
```

- Version de PostgreSQL qui sera installée à partir du dépôt WAPT Deb/RPM.

```
pgsql_version: "11"
```

- Version de CentOS utilisée pour l'adresse du dépôt RPM.

```
centos_version: "centos7"
```

- La valeur par défaut de `launch_postconf` est *True*, elle lance le script de postconfiguration du serveur WAPT en mode silencieux.

```
launch_postconf: True
```

### 73.3.2 Exemple d'un *playbook* Ansible

Voici un exemple d'un *playbook* Ansible.

```
- hosts: srvwapt
  vars_files:
    - vars/main.yml
  roles:
    - tranquilit.waptserver
```

## 73.4 Déployer l'agent Linux WAPT avec Ansible

Pour éviter les erreurs et automatiser le déploiement de votre serveur WAPT, nous fournissons des rôles Ansible pour l'installation du serveur WAPT sur :

- Debian;
- Ubuntu;
- Redhat / CentOS.

Vous pouvez explorer le code source du rôle en [visitant le dépôt Tranquil IT sur Github](#).

### 73.4.1 Pré-requis

- hôtes Debian Linux ou CentOS;
- un sudoer account sur ces machines;
- Ansible 2.8;

### 73.4.2 Installer le rôle Ansible

- Installer le rôle Ansible `tranquilit.waptagent`.

```
ansible-galaxy install tranquilit.waptagent
```

- Pour installer le rôle ailleurs, utilisez la sous-commande `-p` comme ceci.

```
ansible-galaxy install tranquilit.waptagent -p /path/to/role/directory/
```

### 73.4.3 Utiliser le rôle Ansible

- Assurez-vous d'avoir une clé ssh fonctionnelle déployée sur vos hôtes, sinon vous pouvez en générer et la copier comme indiqué ci-dessous.

```
ssh-keygen -t ed25519
ssh-copy-id -i id_ed25519.pub user@computer1.mydomain.lan
ssh user@computer1.mydomain.lan -i id_ed25519.pub
```

- Editer l'inventaire ANsible des machines ( `./hosts` ) et ajouter les machines Linux.

```
[computers]
computer1.mydomain.lan ansible_host=192.168.1.50
computer1.mydomain.lan ansible_host=192.168.1.60
```

— créer un *playbook* avec le contenu suivant dans `./playbooks/deploywaptagent.yml`.

```
- hosts: computers
  roles:
    - { role: tranquil.waptagent }
```

— Assurez-vous que toutes les variables sont correctement définies (voir *variables du wapt-get.ini*).

- `wapt_server_url`;
- `wapt_repo_url`;
- `wapt.crt`.

---

**Important :** Les variables de la configuration est importante car elle va configurer le comportement de WAPT.

vous **devez** remplacer le certificat par défaut avec votre certificat public Code-Signing.

---

— Lancez votre *playbook* avec la commande suivante.

```
ansible-playbook -i ./hosts ./playbooks/deploywaptagent.yml -u user --become --become-method=sudo -K
```

**Félicitations, vous avez installé votre serveur WAPT sur votre serveur Linux !**

### 73.4.4 Paramètres des rôles Ansible

Les variables disponibles sont énumérées ci-dessous, ainsi que les valeurs par défaut (voir `defaults/main.yml`).

### 73.4.5 Variables de l'agent WAPT

— Version de WAPT qui sera installée à partir du dépôt WAPT Deb/RPM.

```
wapt_version: "2.0"
```

— Version de CentOS utilisée pour l'adresse du dépôt RPM.

```
centos_version: "centos7"
```

#### variables du wapt-get.ini

Le paramètre `wapt_server_url` pointe vers votre serveur WAPT et est utilisé par défaut pour le `wapt_repo_url`.

```
wapt_server_url: "https://srvwapt.mydomain.lan"
wapt_repo_url: "{{ wapt_server_url }}/wapt/"
```

Vous pouvez le surcharger comme ceci :

```
wapt_server_url: "https://wapt.landomain.lan"
wapt_repo_url: "https://wapt.otherdomain.com/wapt/"
```



Nom du certificat situé dans le sous-répertoire `files/` du rôle :

```
wapt_cert: "wapt_ca.crt"
```

### 73.4.6 Exemple d'un *playbook* Ansible

Voici un exemple d'un *playbook* Ansible.

```
- hosts: hosts
  vars_files:
    - vars/main.yml
  roles:
    - tranquilit.wptagent
```



---

## Utiliser l'API du serveur WAPT

---

**Note :** Cette documentation ne décrit pas toutes les APIs (Application Protocol Interfaces) disponibles, mais va cependant se concentrer sur les plus utiles.

---

Toutes les URLs disponibles peuvent être trouvées dans `/opt/wapt/waptserver/server.py`.

Les URLs sont formées en utilisant la bonne commande depuis le serveur WAPT ex : `https://srvwapt/command_path`.

---

**Indication :** Cette documentation contient des exemples en code Python ou bien en curl.

---

### 74.1 API V1

#### 74.1.1 /api/v1/hosts

— Récupérer les données enregistrées d'un ou de plusieurs postes.

```
# Args:
#   has_errors (0/1): filter out hosts with packages errors
#   need_upgrade (0/1): filter out hosts with outdated packages
#   groups (csvlist of packages): hosts with packages
#   columns (csvlist of columns):
#   uuid (csvlist of uuid): <uuid1[,uuid2,...]>: filter based on uuid
#   filter (csvlist of field):regular expression: filter based on attributes
#   not_filter (0,1):
#   limit (int): 1000
#   trusted_certs_sha256 (csvlist): filter out machines based on their trusted package certs
```

(suite sur la page suivante)

```
# Returns:
#     result (dict): {'records':[],'files':[]}
#     query:
#         uuid=<uuid>
#     or
#         filter=<csvlist of fields>:regular expression
# """
```

— liste tous les postes. Les paramètres disponibles sont ;

- *reachable*
- *computer\_fqdn* ==> *computer\_name*
- *connected\_ips*
- *mac\_addresses*

Cette exemple montre une requête avec des paramètres :

```
advanced_hosts_wapt = wget('https://%s:%s@%s/api/v1/hosts?columns=reachable,computer_fqdn,
↳connected_ips,mac_addresses&limit=10000' % (wapt_user,wapt_password,wapt_url))
parsed = json.loads(advanced_hosts_wapt)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

Cette exemple est une requête globale :

```
hosts_wapt = wget('https://%s:%s@%s/api/v1/hosts' % (wapt_user,wapt_password,wapt_url))
parsed = json.loads(hosts_wapt)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

**Indication :** Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v1/hosts
```

Celui-ci donne une requête avec un statut joignable, le nom de la machine, ses IP connectées et ses adresses MAC. La limite d'affichage est de 10000 postes

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v1/hosts?columns=reachable,computer_
↳fqdn,connected_ips,mac_addresses&limit=10000
```

## 74.1.2 /api/v1/groups

— récupère tous les paquets groupes. Les groupes peuvent être trouvés avec la section *groupe* dans le paquet.

```
group_wapt = wget('https://%s:%s@%s/api/v1/groups' % (wapt_user,wapt_password,wapt_url))
parsed = json.loads(group_wapt)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

**Indication :** Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v1/groups
```

## /api/v1/host\_data

### dmi

— récupère toutes les informations DMI (Desktop Management Interface) d'un poste :

---

**Note :** ## Récupère des données supplémentaires d'un poste # query : # uuid=<uuid> # field=packages, dmi ou softwares

---

Exemple : récupérer les informations *DMI* du poste ayant l'UUID 14F620FF-DE70-9E5B-996A-B597E8F9B4AD : [https://srvwapt.mydomain.lan/api/v1/host\\_data?uuid=14F620FF-DE70-9E5B-996A-B597E8F9B4AD&field=dmi](https://srvwapt.mydomain.lan/api/v1/host_data?uuid=14F620FF-DE70-9E5B-996A-B597E8F9B4AD&field=dmi)

---

**Note :** le *dmi* n'est pas la seule option disponible. Vous pouvez aussi chercher des informations en utilisant *installed\_packages*, *wsusupdates* ou *installed\_softwares*.

---

```
dmi_host_data_wapt = wget('https://%s:%s@%s/api/v1/host_data?uuid=14F620FF-DE70-9E5B-996A-
↳B597E8F9B4AD&field=dmi' % (wapt_user,wapt_password,wapt_url))
#print(dmi_host_data_wapt)
parsed = json.loads(dmi_host_data_wapt)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

---

**Indication :** Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v1/host_data?uuid=14F620FF-DE70-9E5B-996A-
↳B597E8F9B4AD&field=dmi
```

---

### installed\_packages

L'option *installed\_packages* va lister tous les paquets installés sur un poste en particulier.

```
install_packages_data_wapt = wget('https://%s:%s@%s/api/v1/host_data?uuid=14F620FF-DE70-9E5B-996A-
↳B597E8F9B4AD&field=installed_packages' % (wapt_user,wapt_password,wapt_url))
parsed = json.loads(install_packages_data_wapt)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

---

**Indication :** Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v1/host_data?uuid=14F620FF-DE70-9E5B-996A-
↳B597E8F9B4AD&field=installed_packages
```

---

### installed\_softwares

L'option *installed\_softwares* va lister tous les logiciels installés sur un poste en particulier.

```
install_softwares_data_wapt = wget('https://%s:%s@%/api/v1/host_data?uuid=14F620FF-DE70-9E5B-996A-
↳B597E8F9B4AD&field=installed_softwares' % (wapt_user,wapt_password,wapt_url))
#print(install_softwares_data_wapt)
parsed = json.loads(install_softwares_data_wapt)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

---

**Indication :** Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v1/host_data?uuid=14F620FF-DE70-9E5B-996A-
↳B597E8F9B4AD&field=installed_softwares
```

### wsusupdates

L'option *wsusupdates* va lister toutes les mises à jour installés sur un poste en particulier.

```
wsusupdates_data_wapt = wget('https://%s:%s@%/api/v1/host_data?uuid=14F620FF-DE70-9E5B-996A-
↳B597E8F9B4AD&field=wsusupdates' % (wapt_user,wapt_password,wapt_url))
#print(wsusupdates_data_wapt)
parsed = json.loads(wsusupdates_data_wapt)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

---

**Indication :** Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v1/host_data?uuid=14F620FF-DE70-9E5B-996A-
↳B597E8F9B4AD&field=wsusupdates
```

### 74.1.3 /api/v1/usage\_statistics

Récupère les statistiques d'usage du serveur.

---

**Indication :** Cette API est utile si vous avez plusieurs serveurs WAPT et si vous voulez savoir combien de postes il y a.

```
usage_statistics_wapt = wget('https://%s:%s@%/api/v1/usage_statistics' % (wapt_user,wapt_
↳password,wapt_url))
#print(usage_statistics_wapt)
parsed = json.loads(usage_statistics_wapt)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

---

**Indication :** Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v1/usage_statistics
```

## 74.2 API V2

### 74.2.1 /api/v2/waptagent\_version

Affiche la version du **waptagent.exe** sur le serveur.

```
waptagent_version = wgets('https://%s:%s@%s/api/v2/waptagent_version' % (wapt_user,wapt_password,
↳wapt_url))
parsed = json.loads(waptagent_version)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

**Indication :**

Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v2/waptagent_version
```

## 74.3 API V3

### 74.3.1 /api/v3/packages

Liste les paquets sur le dépôt privé, il récupère le fichier control sur les paquets.

```
packages_wapt = wgets('https://%s:%s@%s/api/v3/packages' % (wapt_user,wapt_password,wapt_url))
parsed = json.loads(packages_wapt)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

**Indication :** Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/packages
```

### 74.3.2 /api/v3/known\_packages

Liste tous les paquets avec l'information *signed\_on*.

```
known_packages_wapt = wgets('https://%s:%s@%s/api/v3/known_packages' % (wapt_user,wapt_password,
↳wapt_url))
parsed = json.loads(known_packages_wapt)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

---

**Indication :** Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/known_packages
```

---

### 74.3.3 /api/v3/trigger\_cancel\_task

Annule une tâche en cours.

```
trigger_cancel_task = wgets('https://%s:%s@%s/api/v3/trigger_cancel_task' % (wapt_user,wapt_
↳password,wapt_url))
parsed = json.loads(trigger_cancel_task)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

### 74.3.4 /api/v3/get\_ad\_ou

Liste les OU vues par les postes et affichées dans la console WAPT.

```
get_ad_ou = wgets('https://%s:%s@%s/api/v3/get_ad_ou' % (wapt_user,wapt_password,wapt_url))
parsed = json.loads(get_ad_ou)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

---

**Indication :** Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/get_ad_ou
```

---

### 74.3.5 /api/v3/get\_ad\_sites

Liste les sites Active Directory.

```
get_ad_sites = wgets('https://%s:%s@%s/api/v3/get_ad_sites' % (wapt_user,wapt_password,wapt_url))
parsed = json.loads(get_ad_sites)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

---

**Indication :** Voici le même exemple avec une simple requête html :



```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/get_ad_sites
```

### 74.3.6 /api/v3/hosts\_for\_package

Liste les postes qui ont un paquet spécifique d'installé `https://srvwapt.mydomain.lan/api/v3/hosts_for_package?package=demo-nompaquet`

```
hosts_for_package = wgets('https://%s:%s@%s/api/v3/hosts_for_package?package=demo-namepackage' %
↳ (wapt_user, wapt_password, wapt_url))
parsed = json.loads(hosts_for_package)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

**Indication :** Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/hosts_for_package?package=demo-namepackage
```

### 74.3.7 /api/v3/host\_tasks\_status

Liste les tâches d'un poste en particulier.

Exemple avec l'uuid d'un poste : `https://srvwapt.mydomain.lan/api/v3/host_tasks_status?uuid=14F620FF-DE70-9E5B-996A-B597E8F9B4AD`

```
host_tasks_status = wgets('https://%s:%s@%s/api/v3/host_tasks_status?uuid=14F620FF-DE70-9E5B-996A-
↳ B597E8F9B4AD' % (wapt_user, wapt_password, wapt_url))
parsed = json.loads(host_tasks_status)
print(json.dumps(parsed, indent=1, sort_keys=True))
```

**Indication :** Voici le même exemple avec une simple requête html :

```
https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/host_tasks_status?uuid=14F620FF-DE70-9E5B-996A-
↳ B597E8F9B4AD
```

**Attention :** Les API ci-après suivent la méthode POST.

### 74.3.8 /api/v3/upload\_packages

---

À faire : Tests

---

### 74.3.9 /api/v3/upload\_hosts

---

À faire : Tests

---

### 74.3.10 /api/v3/change\_password

Change le mot de passe du compte admin [ce compte uniquement]. La requête doit être un dictionnaire python `{}`. Les clés doivent être :

- user;
- password;
- new\_password;

```
curl --insecure -X POST --data-raw '{"user":"admin","password":"OLDPASSWORD","new_password":  
↪ "NEWPASSWORD"}' -H "Content-Type: application/json" "https://admin:OLDPASSWORD@srvwapt/api/v3/  
↪ change_password"
```

### 74.3.11 /api/v3/login

Initialiser une connexion au serveur.

```
curl --insecure -X POST --data-raw '{"user":"admin","password":"MYPASSWORD"}' -H "Content-Type:  
↪ application/json" "https://srvwapt.mydomain.lan/api/v3/login"  
  
{"msg": "Authentication OK", "result": {"edition": "enterprise", "hosts_count": 6, "version": "1.7.4  
↪", "server_domain": "mydomain.lan", "server_uuid": "32464dd6-c261-11e8-87be-cee799b43a00"},  
↪ "success": true, "request_time": 0.03377699851989746}
```

---

**Indication :** Nous pouvons faire une connexion avec un formulaire html plutôt que POST : `https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/get_ad_sites`

---

### 74.3.12 /api/v3/packages\_delete

Supprime un paquet d'une version précise. La requête doit être une liste []. Elle peut prendre plusieurs paquets séparés par des virgules ..

Exemple :

```
curl --insecure -X POST --data-raw '["demo-libreoffice-stable_5.4.6.2-3_all.wapt"]' -H "Content-
↳Type: application/json" "https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/packages_delete"
```

### 74.3.13 /api/v3/reset\_hosts\_sid

Il y a plusieurs possibilités : [https://srvwapt.mydomain.lan/api/v3/reset\\_hosts\\_sid](https://srvwapt.mydomain.lan/api/v3/reset_hosts_sid) va réinitialiser toutes les connexions des postes.

Pour la méthode POST :

La syntaxe est : **--data-raw** : un dictionnaire avec pour clé les uuid et pour valeur l'uuid du poste.

```
curl --insecure -X POST --data-raw '{"uuids":["114F620FF-DE70-9E5B-996A-B597E8F9B4C"]}' -H "Content-
↳Type: application/json" "https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/reset_hosts_sid"

{"msg": "Hosts connection reset launched for 1 host(s)", "result": {}, "success": true, "request_
↳time": null}
```

**Indication** : Si vous voulez plusieurs postes :

```
curl --insecure -X POST --data-raw '{"uuids":["114F620FF-DE70-9E5B-996A-B597E8F9B4C","04F98281-7D37-
↳B35D-8803-8577E0049D15"]}' -H "Content-Type: application/json" "https://admin:MYPASSWORD@srvwapt.
↳mydomain.lan/api/v3/reset_hosts_sid"

{"msg": "Hosts connection reset launched for 2 host(s)", "result": {}, "success": true, "request_
↳time": null}
```

### 74.3.14 /api/v3/trigger\_wakeonlan

Si les postes ont le WakeOnLan d'activé, cette API est utile.

La syntaxe est : **--data-raw** : un dictionnaire avec pour clé les uuid et pour valeur l'uuid du poste.

```
curl --insecure -X POST --data-raw '{"uuids":["04F98281-7D37-B35D-8803-8577E0049D15"]}' -H "Content-
↳Type: application/json" "https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/trigger_wakeonlan"

{"msg": "Wakeonlan packets sent to 1 machines.", "result": [{"computer_fqdn": "win10-1809.mydomain.
↳lan", "mac_addresses": ["7e:c4:f4:9a:87:2d"], "uuid": "04F98281-7D37-B35D-8803-8577E0049D15"}],
↳"success": true, "request_time": null}
```

**Indication** : Si vous voulez plusieurs postes :

```
curl --insecure -X POST --data-raw '{"uuids":["04F98281-7D37-B35D-8803-8577E0049D15","14F620FF-DE70-9E5B-996A-B597E8F9B4AD"]}' -H "Content-Type: application/json" "https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/trigger_wakeonlan"
```

```
{"msg": "Wakeonlan packets sent to 2 machines.", "result": [{"computer_fqdn": "win10-1803.mydomain.lan", "mac_addresses": ["02:4f:25:74:67:71"], "uuid": "14F620FF-DE70-9E5B-996A-B597E8F9B4AD"}, {"computer_fqdn": "win10-1809.ad.alejeune.fr", "mac_addresses": ["7e:c4:f4:9a:87:2d"], "uuid": "04F98281-7D37-B35D-8803-8577E0049D15"}], "success": true, "request_time": null}
```

### 74.3.15 /api/v3/hosts\_delete

""Remove one or several hosts from Server DB and optionnally the host packages

Args:

*uuids (list): list of uuids to delete*  
*filter (csvlist of field:regular expression): filter based on attributes*  
*delete\_packages (bool): delete host's packages*  
*delete\_inventory (bool): delete host's inventory*

Returns:

*result (dict):*  
 ""

Si vous voulez supprimer un poste de l'inventaire :

```
curl --insecure -X POST --data-raw '{"uuids":["04F98281-7D37-B35D-8803-8577E0049D15"],"delete_inventory":"True","delete_packages":"True"}' -H "Content-Type: application/json" "https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/hosts_delete"
```

```
{"msg": "1 files removed from host repository\n1 hosts removed from DB", "result": {"files": ["/var/www/wapt-host/04F98281-7D37-B35D-8803-8577E0049D15.wapt"], "records": [{"computer_fqdn": "win10-1809.mydomain.lan", "uuid": "04F98281-7D37-B35D-8803-8577E0049D15"}]}, "success": true, "request_time": null}
```

Si vous ne voulez pas le supprimer de l'inventaire du serveur :

```
curl --insecure -X POST --data-raw '{"uuids":["04F98281-7D37-B35D-8803-8577E0049D15"],"delete_inventory":"False","delete_packages":"False"}' -H "Content-Type: application/json" "https://admin:MYPASSWORD@srvwapt.mydomain.lan/api/v3/hosts_delete"
```

```
{"msg": "0 files removed from host repository\n1 hosts removed from DB", "result": {"files": [], "records": [{"computer_fqdn": "win10-1809.mydomain.lan", "uuid": "04F98281-7D37-B35D-8803-8577E0049D15"}]}, "success": true, "request_time": null}
```

### 74.3.16 /api/v3/trigger\_host\_action

---

À faire : Tests

---

### 74.3.17 /api/v3/upload\_waptsetup

```
# Upload waptsetup

#Handle the upload of customized waptagent.exe into wapt repository

### DOES NOT WORK
#curl --insecure -X POST -H "Content-Type: multipart/form-data" -F 'data=@waptagent.exe' "https://
↪admin:MYPASSWORD@srvwapt.mydomain.lan/upload_waptsetup"
```

### 74.3.18 /api/v3/ping

Ping va récupérer les informations générales d'un serveur WAPT.

```
# https://srvwapt.mydomain.lan/ping
# Lists WAPT Server informations

ping_wapt = wget('https://%s:%s@s/ping' % (wapt_user,wapt_password,wapt_url))
parsed = json.loads(ping_wapt)
print(json.dumps(parsed, indent=1, sort_keys=True))
```



---

### Contacteur l'éditeur de WAPT

---

Contactez-nous pour plus d'informations :

- **Tranquil IT** : <https://www.tranquil.it/>
- **Twitter** : [https://twitter.com/tranquil\\_it](https://twitter.com/tranquil_it)
- **LinkedIn** : <https://www.linkedin.com/company/tranquil-it>
- **Forum en Français** : <https://forum.tranquil.it/>
- **Forum en Anglais** : <https://www.reddit.com/r/WAPT>
- **Discord** : <https://discord.gg/hFdrqs2C5g>





**Administrateur****Administrateurs****Développeur de Paquet****Développeurs de Paquets**

Un **Administrateur** est un individu pouvant signer des paquets, qu'ils intègrent ou non du code python et les charger sur le dépôt principal.

**Administrateur Local****Administrateurs Locaux**

Un **Administrateur Local** est un utilisateur disposant des droits d'administration locaux sur les postes équipés de WAPT.

**Déploieur de Paquet****Déploieurs de Paquets**

Un **Déploieur de Paquet** est un individu pouvant signer des paquets ne contenant pas de code python (en général les paquets de type *group*, *unit* et *host*) et les charger sur le dépôt principal. Il est typiquement un membre d'une équipe informatique locale qui a une bonne connaissance des besoins des utilisateurs.

**SuperAdmin**

Le **SuperAdmin** est l'*Utilisateur* dont l'identifiant et le mot de passe est défini lors de la post-configuration du serveur WAPT. Dans la version WAPT Discovery, il est l'unique *Administrateur* de WAPT.

**Utilisateur****Utilisateurs**

Un **Utilisateur** est un individu qui utilise une machine équipée de l'agent WAPT (WAPT **Enterprise** et **Discovery**).

**Organisation****Organisations**

L'**Organisation** correspond au périmètre de responsabilité dans lequel est exploitée la solution WAPT.

**ANSSI**

**Agence Nationale de la Sécurité des Systèmes d'Information** est un service français en charge d'assurer la sécurité des informations sensibles de l'État Français et d'une mission de conseil et de soutien aux administrations et aux opérateurs d'importance vitale.

Site internet : <https://www.ssi.gouv.fr/>

### DNS

**Domain Name System** est un service permettant de traduire un nom de domaine en informations de plusieurs types qui y sont associées, notamment en adresses IP de la machine portant ce nom.

### FQDN

**Fully Qualified Domain Name** est un nom de domaine complètement qualifié. C'est la notation complète d'un nom de domaine qui révèle la position absolue de la machine dans l'arborescence DNS en indiquant tous les niveaux supérieurs jusqu'à la racine. Exemple de FQDN : wapt.nantes.pdl.organisation.fr.

### EPEL

**Extra Packages for Enterprise Linux** est un dépôt additionnel pour CentOS et RedHat.

### GPO

Les **Group Policy Objects** ou **Objets de Stratégie de Groupe** sont des objets définissant des stratégies de sécurité dans un environnement Windows. Les stratégies peuvent être définies localement à l'aide de `gpedit.msc` ou définies globalement en domaine Active Directory.

### IDE

**Integrated Development Environment** (environnement de développement intégré) est un ensemble d'outils qui augmente la productivité des développeurs logiciels. Un IDE permet notamment de déboguer ligne par ligne le code source d'un programme, d'éditer, compiler et exécuter dans une seule interface.

### MMC

**Microsoft Management Console** est un gestionnaire de console virtuelle incorporé dans Microsoft Windows, qui sert de conteneur pour des interfaces graphiques de configuration.

### NAT

**Network Address Translation** est un mécanisme qui permet à des machines disposant d'adresses qui font partie d'un intranet de communiquer avec le reste d'Internet en semblant utiliser des adresses externes uniques et routables, au travers d'un routeur.

### Setuphelpers

**Setuphelpers** est une librairie Python écrite spécialement pour WAPT. Elle contient un ensemble de fonctions et de variables utiles au développement de paquets, pour la manipulation de fichiers, création de raccourcis, etc.

### SRV

Le champ **SRV** permet de définir un serveur spécifique pour une application, notamment pour la répartition de charge.

### virtualhost

En informatique, l'**hébergement virtuel** (de l'anglais virtual hosting abrégé **vhost**) est une méthode que les serveurs tels que les serveurs Web utilisent pour accueillir plus d'un nom de domaine sur le même ordinateur, parfois sur la même adresse IP, tout en maintenant une gestion séparée de chacun de ces noms. Cela permet de partager les ressources du serveur, comme la mémoire et le processeur, sans nécessiter que tous les services fournis utilisent le même nom d'hôte.

### waptagent

**waptagent** est l'agent WAPT installé sur chaque ordinateur client.

### waptexit

**waptexit** est une commande qui sera lancé par le script de fermeture de session de Windows (sur les versions Professionnel). La commande lance une mise à jour des paquets en attente.

### waptsetup

**waptsetup** est un logiciel permettant d'installer la console WAPT.

### Websocket

**Websocket** est une couche applicative Web bidirectionnelle permettant la communication client-serveur en utilisant une connexion TCP.

### UUID

Un **UUID** est un identifiant normalisé et réputé unique; ainsi un UUID dans le contexte de WAPT permet d'identifier de manière unique une machine. Pour en savoir plus, suivre [https://en.wikipedia.org/wiki/Universally\\_unique\\_identifier](https://en.wikipedia.org/wiki/Universally_unique_identifier).

**Champ CNAME**

**Champs CNAME**

Un enregistrement **CNAME** ou enregistrement de nom canonique est un type d'enregistrement-ressource dans le Domain Name System (*DNS*) qui spécifie que le nom de domaine est un alias d'un autre nom de domaine canonique.

**Champ A**

**Champs A**

Un **champ A** met en relation un nom (en général le nom physique d'un serveur) avec une IP.

**Autorité de Certification**

Une CA est un tiers de confiance permettant d'authentifier l'identité des correspondants.

**PKI**

**Public Key Infrastructure**, ou Infrastructure à Clés Publiques est un ensemble de composants physiques, de procédures humaines et de logiciels destiné à gérer les clés publiques des utilisateurs d'un système.



---

## Présentation des principes de sécurité

---



Sont documentés ici différents principes avancés de sécurité incorporés dans WAPT.

La lecture de cette documentation n'est pas indispensable pour utiliser WAPT ; elle est cependant recommandée pour vous permettre de mieux comprendre certains choix architecturaux.

### 77.1 Préambule et définitions

**Attention :** Le service WAPT fonctionne en compte système **privilégié**.

**Attention :** A partir de la version 1.5 de WAPT, le répertoire d'installation par défaut devient C:\Program Files (x86)\wapt.

**Indication :** les sous-composantes **waptray**, **waptservice** et **waptexit** de l'agent WAPT peuvent être optionnellement désactivées en fonction du contexte d'usage.

---

## 77.2 Périmètre à sécuriser

Les éléments à sécuriser qui concernent strictement WAPT sont :

- **Le serveur WAPT** (*waptserver*).
- **Les agents WAPT** (*wapt-get*) et ses sous-composantes (*waptray*, *waptservice* et *waptexit*).
- **La console de management** (*waptconsole*).
- **Les communications réseaux** entre ces différentes composantes.

En complément des éléments listés ci-dessus, un exploitant de WAPT devra choisir et suivre une méthodologie adaptée au contexte de son *Organisation* pour :

- Assurer un téléchargement sûr de tous les autres fichiers servant à constituer un paquet WAPT.
- Rédiger le script python `setup.py` d'installation d'un paquet WAPT de telle manière à éviter toute faille de sécurité ou de confidentialité exploitable.
- Gérer de manière sûre les clés privées de signature des paquets.
- Gérer de manière sûre les Autorités de certification et de révocation des certificats SSL et HTTPS.

La gestion sûre de ces éléments complémentaires est exclue du périmètre de cette documentation.

## 77.3 Description des utilisateurs typiques

Les rôles suivants doivent être compris pour évaluer les principes de sécurité présents dans WAPT :

- **Utilisateur**  
Un *Utilisateur* est un individu équipé d'une machine avec l'agent WAPT (**Enterprise** et **Community**).
- **Déploieur de Paquet**  
Un *Déploieur de Paquet* est un individu pouvant signer des paquets ne contenant **PAS** de code python (en général les paquets de type *group*, *unit* et *host*) et les charger sur le dépôt principal (**Enterprise**).
- **Développeur de Paquet**  
Un *Développeur de Paquet\** est un individu pouvant signer des paquets, qu'ils intègrent ou non du code python et les charger sur le dépôt principal (**Enterprise**);

---

**Note :** La distinction entre *Déploieur de Paquet* et *Développeur de Paquet* n'existe que dans la version **Enterprise** de WAPT.

---

- **SuperAdmin**  
Le *SuperAdmin* est un individu avec tous les droits dans WAPT.
- **Administrateur Local**  
Utilisateur disposant des droits d'administration locaux sur les postes équipés de WAPT (**Enterprise** et **Community**);

---

**Note :** En fonction du contexte de la documentation et de la version du produit, un *Administrateur* désignera un *Déploieur de Paquet*, un *Développeur de Paquet* ou bien le *SuperAdmin*.

---

---

**Note :** Les *Utilisateurs* membres du groupe de sécurité Active Directory **waptselfservice** sont considérés comme des *Administrateurs Locaux* du point de vue de la sécurité WAPT.

---

## 77.4 Description des menaces pesant sur les biens sensibles WAPT

Par définition, un bien sensible est une donnée (ou fonction) jugée comme ayant de la valeur pour un attaquant.

Sa valeur est estimée selon des critères de sécurité (aussi appelés besoins de sécurité) :

- Disponibilité;
- Intégrité;
- Confidentialité;
- Authenticité.

Les biens sensibles à protéger sont les suivants :

### 77.4.1 Bien sensible B1 : Les communications

Les communications entre le serveur central et les agents ainsi que les communications entre la console et le serveur sont un bien sensible et doivent être protégées.

---

**Note :** Besoin de sécurité des communications

- Intégrité;
  - Confidentialité;
  - Authenticité.
- 

### 77.4.2 Bien sensible B2 : Les données d'inventaire

Les informations sur l'état de déploiement des paquets, ainsi que configuration matérielle et logicielle des postes clients sont un bien sensible et doivent être protégées.

---

**Note :** Besoin de sécurité des données d'inventaire

- Intégrité;
  - Confidentialité.
- 

### 77.4.3 Bien sensible B3 : Les journaux d'historique

Les journaux générés par WAPT sur le serveur central et les agents sont un bien sensible et doivent être protégés.

---

**Note :** Besoin de sécurité des journaux d'historique

- Disponibilité.
-

#### 77.4.4 Bien sensible B4 : Les valeurs de configuration

Les valeurs de configuration du serveur (clés du serveur https, configuration accès à la base de données, configuration de l'authentification au serveur) sont un bien sensible et doivent être protégés.

---

**Note :** Besoin de sécurité des valeurs de configuration

- Intégrité;
  - Confidentialité.
- 

#### 77.4.5 Bien sensible B5 : Les exécutable WAPT installés sur les postes client

Les exécutable WAPT installés sur les postes client managés (contenu du répertoire wapt incluant les binaires, les dll, les fichiers de configuration et la base de données) sont un bien sensible et doivent être protégés.

---

**Note :** Besoin de sécurité des valeurs de configuration

- Intégrité.
- 

#### 77.4.6 Bien sensible B6 : L'authentification

Les données d'authentification à la console d'administration ainsi que les données d'authentification des agents sur le serveur (clé publique de chaque agent WAPT) sont un bien sensible et doivent être protégés.

---

**Note :** Besoin de sécurité de l'authentification

- Intégrité;
  - Confidentialité.
- 

### 77.5 Description des hypothèses sur l'environnement d'exploitation de WAPT

Par définition, les hypothèses sont des déclarations portant sur le contexte d'emploi de WAPT ou de son environnement.

Les hypothèses suivantes sur l'environnement d'exploitation de WAPT doivent être considérées :

#### 77.5.1 Hypothèse H1 : Les Administrateurs et Déployeurs de Paquet WAPT sont formés

Les *Administrateurs* et les *Développeurs de Paquets* sont formés à l'usage de WAPT. En particulier, ils doivent s'assurer que leurs identifiants et clés de sécurité restent secrets.



### 77.5.2 Hypothèse H2 : Les systèmes subjacents à WAPT sont sains

Les systèmes d'exploitation sur lesquels les agents WAPT s'exécutent mettent en oeuvre des mécanismes de protection adéquats (confinement, contrôle d'accès, etc.) paramétrés et configurés selon les bonnes pratiques.

Les systèmes d'exploitation sont à jour des correctifs en vigueur au moment de l'installation, ils sont sains et exempts de virus, chevaux de Troie, etc.

### 77.5.3 Hypothèse H3 : Les binaires nécessaires au fonctionnement de WAPT sont intègres

Toutes les bibliothèques et outils nécessaires au fonctionnement de WAPT sont considérées saines. A la réception d'une requête par l'agent WAPT, il vérifie que la requête est correctement signée.

### 77.5.4 Hypothèse H4 : Les paquets WAPT sont construits de manière sûre

Il est de la responsabilité de l'*Administrateur* de s'assurer que les fichiers destinés à être intégrés dans des paquets WAPT proviennent de sources sûres et sont en particuliers exempts de virus, chevaux de Troie, etc.

### 77.5.5 Hypothèse H5 : Les Utilisateurs des postes client ne sont pas Administrateurs Locaux

Un *Utilisateur* n'a pas les droits d'administration de son poste de travail. Sinon l'*Utilisateur* est considéré comme un *Administrateur Local*.

En particulier, l'*Utilisateur* n'a pas les droits d'écriture dans le répertoire d'installation du client WAPT.

### 77.5.6 Hypothèse H6 : Les Administrateurs Locaux des postes client sont formés

L'*Administrateur Local* d'un poste client doit être formé à l'exploitation de WAPT, ou à défaut ne pas modifier les fichiers d'installation se trouvant dans le dossier d'installation de WAPT.

## 77.6 Description des menaces pesant sur les biens sensibles WAPT

Par définition, une menace est une action ou un événement susceptible de porter préjudice à la sécurité globale de la machine équipée de WAPT.

Les agents menaçants à considérer pour l'évaluation de sécurité sont les suivants :

- **Entités non-autorisées** : il s'agit d'un attaquant humain ou d'une entité qui interagit avec WAPT mais qui ne dispose pas d'un accès légitime à celui-ci.

---

**Note** : Les *Administrateurs* et les *Administrateurs Locaux* ne sont pas considérés comme des attaquants.

---

Les menaces qui portent sur les biens sensibles WAPT définis ci-dessus sont les suivantes :

### 77.6.1 Menace M1 : Installation d'un logiciel malveillant par une entité non-autorisée

Cette menace correspond à un attaquant qui parviendrait à utiliser une composante de l'agent WAPT pour installer une application malveillante de façon pérenne, ou pour désinstaller ou désactiver une composante de sécurité du poste sur lequel l'agent WAPT est installé.

### 77.6.2 Menace M2 : Altération de valeurs de configuration par une entité non-autorisée

Cette menace correspond à un attaquant qui parviendrait à modifier ou à supprimer le paramétrage d'un élément de WAPT défini par un *Administrateur* légitime de WAPT.

### 77.6.3 Menace M3 : Accès illégitime par une entité non-autorisée

Cette menace correspond à un attaquant qui parviendrait à récupérer les données d'authentification d'un *Administrateur*, à contourner le mécanisme d'authentification de manière à accéder ou à altérer un bien sensible stocké sur le serveur. Elle correspond également à un attaquant qui parviendrait à se faire passer pour un agent WAPT.

### 77.6.4 Menace M4 : Écoute du réseau par une entité non-autorisée

Cette menace correspond à un attaquant qui parviendrait à intercepter et prendre connaissance des communications réseaux entre les agents et le serveur hébergeant WAPT.

### 77.6.5 Menace M5 : Altération du trafic réseau par une entité non-autorisée (Type *Man In the Middle*)

Cette menace correspond à un attaquant qui parviendrait à modifier les communications réseaux entre les agents et le serveur hébergeant WAPT ou les communications réseau entre la console et le serveur WAPT.

## 77.7 Description des fonctions de sécurité de WAPT

Par définition, les fonctions de sécurité sont l'ensemble des mesures techniques et mécanismes mis en œuvre pour protéger de façon proportionnée les biens sensibles contre les menaces identifiées.

### 77.7.1 Fonction de sécurité F1 : Authentification des contrôles d'accès

**Fonction de sécurité F1A : Authentification d'une machine lors de son enregistrement initial dans la base de données WAPT**

Nouveau dans la version 1.5.

---

**Note :** risques traités

- L'inscription d'une machine illégitime dans la base de données.
  - Une attaque par déni de service par surcharge de la base de données.
  - Eviter l'enregistrement d'un inventaire falsifié dans la base de données.
-

## Solution mise en place

Pour exister dans la base de données et ainsi apparaître dans la console WAPT, une machine doit s'enregistrer auprès du serveur WAPT avec une commande **register**.

La commande **register** peut être exécutée automatiquement lors de l'installation ou de la mise à jour de l'agent WAPT si la machine est correctement enregistrée avec un compte machine Kerberos dans le domaine Active Directory de l'*Organisation*.

Si la machine ne présente pas au serveur WAPT un ticket Kerberos valide, alors la commande **register** échoue ;

---

**Note :** La méthode avec Kerberos assume que le serveur Active Directory répond au moment du **register**.

---

## Fonction de sécurité F1B : Vérification des certificats HTTPS du serveur par les clients WAPT

Nouveau dans la version 1.5.

---

**Note :** risques traités (notamment MITM) :

- L'envoi d'informations sensibles à un serveur WAPT illégitime et non-autorisé.
  - La récupération d'informations sensibles par une entité non-autorisée.
  - L'affichage d'informations falsifiées dans la console de l'*Administrateur*.
  - Une mauvaise date est envoyée lors d'une requête HEAD (demande de modification de date de fichier) empêchant les futures mise à jours.
  - Envoyer le mot de passe de la console à un serveur WAPT illégitime et non-autorisé.
- 

## Solution mise en place

Pour fonctionner correctement en version sécurisée :

- Une option de vérification du certificat HTTPS serveur est introduite dans le fichier C:\Program Files (x86)\wapt\wapt-get.ini des agents WAPT qui **force la vérification du certificat serveur par les agents WAPT** .
- Une option de vérification du certificat HTTPS serveur est introduite dans le fichier C:\Program Files (x86)\wapt\wapt-get.ini des agents WAPT qui **force la vérification du certificat serveur par les agents WAPT**.

L'implémentation technique peut être basée sur deux méthodes :

- Utiliser un utilitaire de vérification de certificat implémenté dans la configuration du service **Nginx** du serveur WAPT ; cette méthode est généralement fournie par une *Autorité de Certification* validée pour votre réseau.
- Utiliser la méthode d'*épinglage de certificat* qui consiste à fournir à l'agent WAPT une liste de certificats de confiance qui sera stockée et maintenue dans le dossier C:\Program Files (x86)\wapt\ssl\server .

## Fonction de sécurité F1C : Aucun port n'écoute sur les agents WAPT

Nouveau dans la version 1.5.

---

**Note :** risques traités

- Une entité non-autorisée utilise un port ouvert à mauvais escient.
-

### Solution mise en place

Les connexions vers le serveur WAPT sont exclusivement initiées pas les clients, et les différentes actions instantanées (**update / upgrade / install ...**) passent au travers d'une connexion permanente par une Websocket initiée par l'agent WAPT.

---

**Note :** si HTTPS est activé, l'agent vérifie que la Websocket s'établit bien avec le bon serveur.

---

### Fonction de sécurité F1D : Signature des remontées d'inventaire

Nouveau dans la version 1.3.12.13.

---

**Note :** risques traités

- Une entité non-autorisée envoie un inventaire falsifié d'une machine existante dans la base de données WAPT.
- 

### Solution mise en place

- Au premier **register**, chaque machine crée un couple clé privée / certificat public dans le répertoire C:\Program Files (x86)\wapt\private accessible en lecture uniquement aux *Administrateurs Locaux*. Une fois la machine enregistrée, la clé publique est envoyée au serveur WAPT.
- Lors d'une mise à jour de l'inventaire, le nouvel inventaire est envoyé signé avec la clé privée de la machine et déchiffré par la clé publique enregistrée dans la base de données.
- Le serveur refusera de valider tout inventaire signé avec une mauvaise clé.

### Fonction de sécurité F1E : Vérification des droits avant l'exécution de certaines actions WAPT

---

**Note :** risques traités

- Eviter l'exécution de tâches sensibles par des entités non-autorisées.
- 

### Solution mise en place

Les *Utilisateurs* interagissent avec WAPT au travers des interfaces WAPT (**wapt-get** en ligne de commande, **wapttray**, **waptexit**, **waptselfservice**).

Les interfaces peuvent ensuite déléguer l'exécution des tâches souhaitées au service WAPT local fonctionnant en compte système.

Les actions qui enclenchent des modifications listées ci-dessous ne nécessitent pas d'authentification auprès du service WAPT :

- **wapt-get update** (mettre à jour la liste des paquets disponibles).
- **wapt-get upgrade** (lancer l'installation des mises à jour en attente).
- **wapt-get download-upgrade** (télécharger les mises à jour en attente).
- **wapt-get clean** (supprimer des paquets restés en cache après installation).
- stopper n'importe quelle tâche WAPT en cours.
- stopper / relancer le service WAPT.

Les autres actions nécessitent que l'*Utilisateur* s'authentifie et que son compte appartienne au groupe de sécurité Active Directory **waptselfservice** ou que l'*Utilisateur* soit *Administrateur Local*, exemple d'action :

- `wapt-get install` : ordonner à l'agent WAPT d'installer un paquet WAPT marqué **MISSING** sur la machine.
- `wapt-get remove` : ordonner à l'agent WAPT de supprimer un paquet WAPT.
- `wapt-get forget` : ordonner à l'agent WAPT d'oublier l'existence d'un paquet WAPT installé sur la machine sans le dés-installer.

## 77.7.2 Fonction de sécurité F2 : Protection de l'intégrité du processus d'installation des paquets WAPT

### Fonction de sécurité F2A : Signature des paquets WAPT

---

**Note** : risques traités

- Pour éviter qu'une entité non-autorisée modifie le contenu ou le comportement d'un paquet WAPT.
- 

#### Solution mise en place

- Quand un *Administrateur* ou un *Développeur de Paquet* construit un paquet WAPT, un fichier `WAPTmanifest.sha256` est créé qui liste les sommes de contrôle de tous les fichiers du paquet.
- Un fichier `signature.sha256` **chiffré** avec la clé privée est ensuite créé dans le dossier WAPT, il contient la somme de contrôle du fichier `WAPTmanifest.sha256`.
- L'ensemble est archivé avec l'extension `.wapt`.
- Quand un agent WAPT télécharge un paquet WAPT, l'agent vérifie que le fichier `signature.sha256` a été signé avec la clé privée qui correspond au certificat présent dans le dossier WAPT.
- L'agent WAPT vérifie ensuite que le certificat (ou la chaîne de certificat) `certificate.crt` a bien été signé avec une clé privée correspondant à un des certificats présents dans le dossier `C:\Program Files (x86)\wapt\ssl`.
- L'agent WAPT fait ensuite la somme de contrôle de tous les fichiers du paquet (excepté les fichiers `signature.sha256` et `certificate.crt`) et vérifie que cela correspond au fichier `WAPTmanifest.sha256` contenu dans le paquet.
- Si l'une de ces étapes n'est pas validée, alors cela signifie qu'un fichier a été modifié/ ajouté/ supprimé. Alors, l'exécution du `setup.py` est annulée.
- Le paquet en défaut est ensuite supprimé du cache local ; l'évènement est rapporté dans les log de l'agent.

---

**Note** : Depuis la version 1.5, le fichier `manifest` est passée de `sha1` à `sha256`.

---

### Fonction de sécurité F2B : Signature des attributs du fichier *control*

Nouveau dans la version 1.4.

---

**Note** : risques traités

- Une entité non-autorisée modifie des dépendances WAPT sur la machine en falsifiant le fichier `https://waptserver/wapt/Packages`.
-

### Solution mise en place

Lors de la signature d'un paquet WAPT, les attributs sensibles du paquet sont listés dans l'attribut **signed\_attributes**.

---

**Note :** Exemple d'une liste *signed\_attributes* :

*package, version, architecture, section, priority, maintainer, description, depends, conflicts, maturity, locale, min\_os\_version, max\_os\_version, min\_wapt\_version, sources, installed\_size, signer, signer\_fingerprint, signature\_date, signed\_attributes,*

---

Les attributs listés dans *signed\_attributes* sont signés avec la clé privée de l'*Administrateur* et la signature est stockée dans l'attribut *signature* du fichier **control**.

Le certificat associé à cette clé privée est stocké dans le fichier `WAPT\certificate.crt` à l'intérieur du paquet WAPT.

Sur le serveur WAPT, lors de l'opération **wapt-scanpackages** (déclenchée par un ajout ou suppression de paquet), l'index Packages des paquets est régénéré.

Le serveur WAPT extrait de chaque paquet le certificat du signataire et l'ajoute dans le fichier ZIP Packages, dans le répertoire `ssl`. Chaque certificat est nommé avec sa fingerprint encodée en hexadécimal.

Lorsque le client WAPT effectue un **update** (mise à jour des paquets disponibles), il télécharge le fichier index Packages, qui contient à la fois les attributs signés de tous les paquets et les certificats des signataires.

Si le certificat du signataire des attributs d'un paquet est approuvé (ce qui signifie que ce certificat est signé par une *Autorité de Certification* ou que le certificat lui-même est de confiance), **ET** que le certificat du signataire peut vérifier la signature des attributs, le paquet est ajouté à l'index des paquets disponibles, sinon il est ignoré.

### Fonction de sécurité F2C : Restriction d'accès au répertoire d'installation de l'agent WAPT

---

**Note :** risques traités

- Une entité non-autorisée modifie le comportement de l'agent WAPT.
- 

Le répertoire d'installation `C:\Program Files (x86)\wapt` est accessible en lecture et modification :

- Aux *Administrateurs Locaux* à travers un accès local au répertoire d'installation de l'agent WAPT.
- Aux *Administrateurs* à travers le mécanisme de déploiement des mises à jour de l'agent WAPT.

Ni les *Développeurs de Paquets*, ni les *Utilisateurs* n'ont d'accès en écriture au répertoire d'installation de l'agent WAPT.

### Fonction de sécurité F2D : Restriction totale d'accès au répertoire de stockage du couple clé privé / certificat de signature d'inventaire

---

**Note :** risques traités

- Une entité non-autorisée falsifie une remontée d'inventaire.
  - Une entité non-autorisée usurpe l'identité d'une machine avec WAPT.
- 

Aucun droit d'accès au répertoire `C:\Program Files (x86)\wapt\private` n'est accordé à aucun *Utilisateur*, quel qu'il soit. Seul l'agent WAPT a accès en lecture et écriture à ce répertoire.

---

**Note :** Le stockage du couple clé privée / certificat découle d'un choix technique qui consiste à dire que la machine détient seule toutes les informations qui la concernent.

---

### 77.7.3 Fonction de sécurité F3 : Sécurisation des communications entre les différents composants WAPT

#### Fonction de sécurité F3A : Signature des requêtes envoyées aux agents WAPT

Nouveau dans la version 1.5.

---

**Note :** risques traités

- Une entité non-autorisée envoie des requêtes falsifiées aux agents WAPT.
- 

#### Solution mise en place

Les commandes ci-dessous sont signées par le serveur WAPT avant d'être envoyées au travers de la Websocket à l'agent WAPT destinataire de la commande :

- `wapt-get install` : ordonner à l'agent WAPT d'installer un paquet WAPT marqué **MISSING** sur la machine.
- `wapt-get remove` : ordonner à l'agent WAPT de supprimer un paquet WAPT.
- `wapt-get forget` : ordonner à l'agent WAPT d'oublier l'existence d'un paquet WAPT installé sur la machine sans le dés-installer.
- `wapt-get update-status` : ordonner à l'agent WAPT de renvoyer l'état de son inventaire actuel au serveur WAPT.
- `wapt-get upgrade` : ordonner à l'agent WAPT d'exécuter les paquets marqués **NEED UPGRADE**.
- `wapt-get update` : ordonner à l'agent WAPT de mettre à jour la liste des paquets disponibles.

L'ensemble des attributs de cette requête sont signés :

- L'*UUID* du poste;
- L'action (ex : **install**);
- Les arguments (ex : `tis-firefox`);
- L'horodatage des requêtes.

Le certificate associé à la signature est également passé :

- A la réception d'une requête par l'agent WAPT, il vérifie que la requête est correctement signée.
- L'agent vérifie ensuite que la date fournie en argument ne dépasse pas une minute de décalage.
- Enfin, l'agent vérifiera enfin que le certificat associé est autorisé à lancer des commandes.

## 77.8 Présentation des processus cryptographiques

Date	janv. 09, 2024
Rédacteur	Hubert TOUVET
Applicable pour WAPT	>= 1.5.0.17
Version du document	1.5.0.17-0

- *Répertoires et fichiers référencés dans ce document*
- *Définition des Acteurs*
- *Synthèse des modules crypto mis en oeuvre par la solution WAPT*
- *Gestion des clés et des certificats de l'Administrateur*
  - *Validité du certificat de l'Administrateur*
  - *Autoriser le certificat de l'Administrateur à signer un paquet*
- *Gérer les clés et certificats du Client WAPT*
  - *Émission initiale et mise à jour du certificat du client WAPT*
  - *Déployer les certificats d'autorité pour vérifier les paquets et les actions sur les clients*
  - *Déployer les certificats d'autorité pour la communication HTTPS entre les clients WAPT et le serveur WAPT*
- *Communications HTTPS entre les clients WAPT et les dépôts WAPT*
  - *Déployer des certificats d'autorité*
  - *Communications Websockets entre les clients WAPT et le serveur WAPT*
- *Communications entre la console WAPT et le serveur WAPT*
  - *Déployer des certificats d'autorité*
  - *Déployer des certificats d'autorité pour vérifier les paquets importés dans le dépôt local*
- *Processus de signature d'un paquet*
  - *Paramètres initiaux*
  - *Signature des attributs du fichier control*
  - *Signature des fichiers du paquet*
- *Vérifier la signature des attributs d'un paquet*
- *Vérifier la signature d'un paquet*
- *Signature d'une action immédiate*
  - *Processus de signature pour des actions immédiates*
  - *Vérifier la signature d'une action immédiate*
- *Vérification du téléchargement complet d'un paquet*

Les processus cryptographiques sont utilisés dans les activités suivantes :

- Signature et vérification des **fichiers contenus dans un paquet**.
- Signature et vérification des **attributs d'un paquet**.
- Signature et vérification des **actions immédiates** sur les client WAPT.
- Signature des inventaires et **statut des clients WAPT**.
- Authentification de la connexion Websockets du client WAPT sur le serveur.
- Communication https entre les clients WAPT et le serveur WAPT.
- Communications HTTPS entre la console WAPT et le serveur WAPT.
- Communications HTTPS entre les clients WAPT et les dépôts WAPT.

### 77.8.1 Répertoires et fichiers référencés dans ce document

- <WAPT> : répertoire d'installation de WAPT. Par défaut %Program Files (x86)%WAPT.
- <WAPT>wapt-get.ini : fichier de configuration du client WAPT (**wapt-get** et **waptservice**).
- <WAPT>ssl : répertoire par défaut pour les certificats de confiance des paquets et actions.
- <WAPT>sslserver : répertoire par défaut pour stocker les certificats https du serveur (pinning).
- <WAPT>\private : répertoire par défaut pour les certificats permettant de signer l'inventaire et les connexions Websocket.
- %LOCALAPPDATA%waptconsolewaptconsole.ini : fichier de configuration de la console et des actions de développement de l'outil **wapt-get**.
- %appdata%waptconsolessl : répertoire par défaut pour les certificats de confiance pour l'import de paquets depuis un dépôt externe (c.à.d. les *modèles de paquets*).



## 77.8.2 Définition des Acteurs

### — Organisation

Une Organisation est le périmètre de responsabilité dans lequel est exploitée la solution WAPT.

### — Autorité de Certification

Un certificat d'Autorité est l'entité qui détient les clés qui ont signé les certificats des *Développeurs de Paquets*, des *Déploieurs de Paquets* et des serveurs HTTPS.

### — Administrateurs

Les Administrateurs sont en possession d'une clé RSA personnelle et d'un certificat signé par l'*Autorité de Certification* de l'*Organisation* ; ils ont aussi un identifiant et un mot de passe pour accéder à la console WAPT.

### — Postes clients WAPT

Les clients WAPT sont l'ensemble des appareils que les *Administrateurs* de l'*Organisation* peuvent gérer avec WAPT. Les clients **peuvent être ou non un membre** du domaine Active Directory de l'*Organisation*.

### — Serveur WAPT

Le serveur WAPT est un serveur Linux / Nginx / PostgreSQL de l'*Organisation* qui gère l'inventaire et le status des Postes clients WAPT.

Par défaut, le serveur WAPT joue également le rôle de dépôt WAPT interne. Le serveur WAPT a un compte ordinateur dans l'Active Directory de l'*Organisation*.

### — Dépôts WAPT internes

Les dépôts internet WAPT sont un ou plusieurs serveur Linux / Nginx qui diffusent aux Postes clients WAPT en HTTPS des paquets WAPT signés.

### — Dépôts WAPT externes

Les dépôts WAPT externe sont des dépôts WAPT publics que les *Développeurs de Paquets* peuvent utiliser pour importer des paquets conçus par d'autres *Organisations*, sous condition d'en vérifier l'adéquation aux normes internes de sûreté et de sécurité ;

### — Serveur Active Directory

Le serveur Active Directory gérant le domaine AD de l'*Organisation* ;

## 77.8.3 Synthèse des modules crypto mis en oeuvre par la solution WAPT

---

À faire : sfonteneau

---

Coté client WAPT (WAPT 1.5.0.12) :

- module *ssl* standard de **Python 2.7.13** lié sur **OpenSSL 1.0.2j 26 Sep 2016** pour les connexions https entre les clients WAPT et serveur WAPT.
- **cryptography==1.9** lié sur **openssl 1.1.0f** pour toutes les opérations crypto RSA, génération de clés, de certificat X509, de signature et vérification.
- **kerberos-sspi==0.2** et **requests-kerberos==0.11.0** pour l'authentification du client WAPT lors de son enregistrement initial sur le serveur.
- **pyOpenSSL==17.0.0** : pour récupérer la chaîne de certificats du serveur WAPT.
- **certifi==2017.4.17** : base pour les certificats d'Autorité Racine.
- **dll OpenSSL 1.0.2l** pour la partie waptcommon.pas écrite avec la bibliothèque FPC Indy et la classe TIdSSLIOHandlerSocketOpenSSL.

Coté serveur WAPT :

- **nginx/1.10.2** : configurée pour TLS1.2, chiffre "EECDH+AESGCM :EDH+AESGCM :AES256+EECDH :AES256+EDH".
- module *ssl* standard de **python 2.7.5** lié sur **OpenSSL 1.0.1e-fips 11 Feb 2013**.
- **cryptography==1.9** lié sur **OpenSSL 1.0.1e-fips 11 Feb 2013** pour toutes les opérations crypto RSA, X509, signature et vérification.

## 77.8.4 Gestion des clés et des certificat de l'Administrateur

Les paquets et actions de l'Administrateur sont signés pour n'autoriser que les Administrateurs de confiance à intervenir sur les postes.

L'Administrateur de la solution WAPT a en sa possession :

- Une clé privée RSA de 2048 bits chiffrée par l'algorithme aes-256-cbc.
- Un certificat X509 signé par une Autorité de Certification approuvée par l'Organisation.

---

**Note :** Le processus d'émission de ces clés, la signature du certificat, la distribution et la révocation sont à la charge de l'Organisation utilisant WAPT et sortent donc du périmètre fonctionnel de WAPT.

Cependant, pour facilement tester la solution, WAPT propose une fonction pour générer une clé RSA et un certificat X509 :

- La clé RSA générée est de 2048 bits, chiffrée par l'algorithme aes-256-cbc et encodée en format PEM avec l'extension `.pem`.
  - Le certificat est soit autosigné, soit signé par une Autorité de Confiance dont on a à disposition la clé et le certificat en format PEM.
  - Si le certificat est autosigné, son attribut `KeyUsage` comporte le flag `keyCertSign`.
  - Si l'Administrateur est habilité par l'Organisation à signer des paquets contenant du code python (présence du fichier `setup.py`), l'attribut du certificat `extendedKeyUsage` comporte le flag **CodeSigning**.
  - Le certificat X509 est encodé et remis à l'Administrateur en format PEM avec l'extension `.crt`.
- 

### Validité du certificat de l'Administrateur

Jusqu'à la version 1.5.0.12 incluse, Le client WAPT ne gère pas la vérification de la révocation du certificat de l'Administrateur lors du processus de vérification des paquets, attributs et actions de l'Administrateur.

Il ne vérifie que les dates de validité (attributs `notValidBefore` / `notValidAfter`). Le certificat est valide si (`Now`  $\geq$  `notValidBefore` et `Now`  $\leq$  `notValidAfter`).

### Autoriser le certificat de l'Administrateur à signer un paquet

Le certificat utilisé par la console WAPT pour signer les paquets et actions est défini avec le paramètre `personal_certificate_path` de la section `[global]` du fichier `%LOCALAPPDATA%\waptconsole\waptconsole.ini`.

WAPT demande à l'Administrateur son mot de passe pour permettre de rechercher la clé privée (encodée au format PEM) correspondant au certificat parmi les fichiers `.pem` du répertoire contenant les certificats.

Lors de la signature de paquet, WAPT refusera le certificat si le paquet contient un fichier `setup.py` et que le certificat n'est pas de type `CodeSigning`.

## 77.8.5 Gérer les clés et certificats du Client WAPT

Le client WAPT (**waptservice**) utilise des clés RSA et un certificat X509 pour interagir avec le serveur WAPT.

Le certificat du client WAPT est utilisé dans les situations suivantes :

- Lors de la mise à jour du statut du poste sur le serveur. (**update\_server\_status**) : **signature des informations**.
- Lors de la connexion Websocket du poste vers le serveur (**waptservice**) : **signature de l'UUID du poste**.

## Émission initiale et mise à jour du certificat du client WAPT

- A l'issu du processus d'installation de l'agent WAPT sur le Poste client, l'agent WAPT s'enregistre automatiquement auprès du serveur WAPT en émettant une requête https authentifiée par Kerberos qui utilise le TGT du compte machine.

L'agent WAPT utilise les API kerberos de Windows en s'appuyant sur les modules python **kerberos-sspi** et **requests-kerberos**.

---

**Note :** cette procédure fonctionne si et seulement si le Poste client est joint au domaine Windows pour lequel le serveur WAPT est configuré.

---

Si la clé et les certificats n'ont pas encore été générés, ou s'ils ne correspondent pas au *FQDN* actuel de la machine, l'agent WAPT génère une clé RSA et un certificat X509 autosigné avec les paramètres suivants :

- La clé est de type RSA 2048 bits encodée en PEM et stockée dans le fichier `<WAPT>\private<fqdn du poste>.pem`.
- Le certificat généré a les attributs suivants :
  - `Subject.COMMON_NAME*` = `<FQDN e l'appareil>`.
  - `Subject.ORGANIZATIONAL_UNIT_NAME` = nom de l'*Organisation* enregistré la base de registre du client Windows.
  - `SubjectAlternativeName.DNSName` = `<device FQDN>`.
  - `BasicConstraint.CA` = `True`.
  - `Validity` = 10 ans.
  - `Serialnumber` = aléatoire ;
- Le certificat est sauvegardé dans le fichier `<WAPT>private<device FQDN>.crt`.

---

**Note :** Seuls le compte machine et les *Administrateurs Locaux* ont accès au répertoire `<WAPT>\private` car des ACL spécifiques sont appliquées à l'installation de l'agent WAPT sur le poste.

---

- L'inventaire ou les mises à jour de status du client sont envoyés au serveur WAPT par requête https POST ;
- On authentifie la requête https POST en ajoutant deux headers http spécifiques :
  - *X-Signature* :
    - Encodage en JSON des informations BLOB d'inventaire ou de status.
    - signature du json avec la clé privée du client WAPT : hachage *sha256* et padding *PKCS#1 v1.5*.
    - encodage de la signature en *base64*.
  - *X-Signer* : `Subject.COMMON_NAME` ou UUID du client WAPT.
- Après avoir initialement authentifié le client WAPT avec Kerberos, le serveur reçoit le certificat envoyé par le client et il le stocke dans son inventaire, dans la table *hosts* (champ *host\_certificate* en format PEM).

---

**Note :** Si le poste client WAPT est renommé, la paire de clés et le certificat sont recréés.

Lors de la tentative de mise à jour de status du client vers le serveur, la requête POST sera refusée, car la machine est enregistrée dans la base de données avec un autre certificat.

La machine tentera alors de se ré-enregistrer (**register**) avec authentification kerberos ; ainsi le nouveau certificat sera enregistré dans la base de données.

---

### Déployer les certificats d'autorité pour vérifier les paquets et les actions sur les clients

Les certificats formattés en PEM sont stockés dans des fichiers avec des extensions `.crt` or `.pem` dans le répertoire défini avec le paramètre `public_certs_dir` dans la fichier `<WAPT>wapt-get.ini`. Ils sont réputés pour être des **certificats d'autorité**.

Ce paramètre `public_certs_dir` est initialisé par défaut à `<WAPT>ssl`.

Le déploiement de ces certificats d'autorité est effectué lors de l'installation initiale de l'agent WAPT par l'installateur.

Depuis la console, l'*Administrateur* compile un installateur personnalisé en vue de son déploiement par *GPO* sur les Postes clients.

La console WAPT incorpore dans cet installateur les certificats présents dans le répertoire `<WAPT>\ssl` du poste depuis lequel l'installateur est compilé.

L'*Administrateur* doit s'assurer d'enregistrer dans `<WAPT>ssl` uniquement les certificats d'autorité nécessaires avant de lancer la compilation de l'agent.

Le déploiement ou la mise à jour de certificats de l'*Autorité de Certification* pour la vérification des paquets et actions peuvent être également assurés à postériori par une *GPO Active Directory* ou par un paquet WAPT.

### Déployer les certificats d'autorité pour la communication HTTPS entre les clients WAPT et le serveur WAPT

Le service WAPT ainsi que l'outil en ligne de commande `wapt-get` communiquent avec le serveur WAPT pour envoyer l'inventaire (**register**) et le statut de déploiement des paquets (**update-status**).

Ces deux types de connexions vérifient le certificat https du serveur.

Paramètre `verify_cert` de la section `[global]` du fichier `<WAPT>wapt-get.ini` :

- `verify_cert = 1`  
cette méthode ne fonctionnera bien que si le serveur https est configuré pour renvoyer son certificat et les certificats intermédiaires à l'initialisation de communication TLS.
- `verify_cert = <chemin vers fichier .pem>`  
vérifie le certificat du serveur https en utilisant le bundle de certificats indiqué. Tous les certificats de CA intermédiaires et root doivent être rassemblés dans un fichier au format `.pem` ;
- `verify_cert = 0`  
ne pas vérifier le certificat du serveur https ;

Conventionnellement, on stocke le bundle de l'*Autorité de Certification* approuvées dans le répertoire `<WAPT>sslserver`.

La console WAPT comporte une fonction pour faciliter la récupération initiale de la chaîne de certificats du serveur et pour la stocker au format `.pem` dans le fichier `<WAPT>sslserver<FQDN serveur>.pem`.

Il est de la responsabilité de l'*Administrateur* de s'assurer que la chaîne ainsi récupérée est authentique.

Lors de la compilation de l'installateur de l'agent WAPT, les certificats ou le bundle de certificats sont intégrés dans l'installateur.

Lors du déploiement de l'installateur sur les clients WAPT, le bundle est copié dans `<WAPT>sslserver` et le paramètre `verify_cert` de la section `[global]` du fichier `<WAPT>wapt-get.ini` est renseigné pour désigner le bundle.

## 77.8.6 Communications HTTPS entre les clients WAPT et les dépôts WAPT

### Déployer des certificats d'autorité

Les connexions HTTPS de l'agent WAPT vers le dépôt principal utilisent les mêmes méthodes que les communications entre l'agent WAPT et le serveur WAPT.

L'agent WAPT utilise le même bundle de certificats pour communiquer en HTTPS avec le dépôt principal, avec le serveur WAPT, et avec les dépôts secondaires.

La connexion https est mise en œuvre par les modules python **requests**, **urllib3** et **ssl**.

Le certificat transmis par le serveur HTTPS du dépôt est vérifié par le module **urllib3.contrib.pysopenssl.PyOpenSSLContext** et **urllib3.util.ssl\_wrap\_socket**.

### Communications Websockets entre les clients WAPT et le serveur WAPT

Pour permettre des actions immédiates sur les clients WAPT, le service WAPT déployé sur les clients tente d'établir et de maintenir une connexion WebSocket vers le serveur WAPT.

Cette connexion s'effectue sur une connexion chiffrée avec le protocole TLS et utilise côté client le même bundle de certificat que la connexion HTTPS Client vers Serveur WAPT.

## 77.8.7 Communications entre la console WAPT et le serveur WAPT

### Déployer des certificats d'autorité

Paramètre *verify\_cert* de la section [global] du fichier %LOCALAPPDATA%\waptconsolewaptconsole.ini :

- *verify\_cert* = 1  
cette méthode ne fonctionnera bien que si le serveur https est configuré pour renvoyer son certificat et les certificats intermédiaires à l'initialisation de communication TLS.
- *verify\_cert* = <chemin vers fichier .pem>  
vérifie le certificat du serveur https en utilisant le bundle de certificats indiqué. Tous les certificats de CA intermédiaires et root doivent être rassemblés dans un fichier au format .pem ;
- *verify\_cert* = 0  
ne pas vérifier le certificat du serveur https ;

Conventionnellement, on stocke le bundle de l'*Autorité de Certification* approuvées dans le répertoire <WAPT>sslserver.

La console WAPT comporte une fonction pour faciliter la récupération initiale de la chaîne de certificats du serveur et la stocker au format .pem dans le fichier <WAPT>sslserver<FQDN serveur>.

Il est de la responsabilité de l'*Administrateur* de s'assurer que la chaîne ainsi récupérée est authentique.

Il est également possible de récupérer la chaîne de certificats du serveur et de renseigner le paramètre *verify\_cert* avec la commande **wapt-get enable-check-certificate**.

### Déployer des certificats d'autorité pour vérifier les paquets importés dans le dépôt local

Dans la console WAPT / onglet *Dépôt privé*, un bouton *Importer depuis internet* permet de télécharger un paquet depuis un dépôt externe dont l'URL est fournie par le paramètre *repo\_url* de la section [wapt\_templates] du fichier %LOCALAPPDATA%waptconsolewaptconsole.ini.

Une case à cocher *Vérifier la signature de paquet* permet de s'assurer que le paquet est signé avec un certificat provenant d'une Autorité de confiance.

Les certificats d'autorité présents dans le répertoire désigné par le paramètre *public\_certs\_dir* de la section [wapt\_templates] du fichier %LOCALAPPDATA%waptconsolewaptconsole.ini sont réputés de confiance.

Si le paramètre n'est pas mentionné explicitement, il est initialisé à %appdata%waptconsolessl.

Ce répertoire n'est pas automatiquement rempli par WAPT. Il est de la responsabilité de l'Administrateur de copier/coller les fichiers PEM d'autres Administrateurs ou les certificats des Autorités de Certification.

Les certificats d'autorité sont encodés en format PEM et stockés dans des fichiers avec l'extension *.pem* ou *.crt*. On peut stocker plusieurs certificats dans chaque fichier *.crt* ou *.pem*.

Il n'est pas nécessaire d'avoir la chaîne complète de certificats, WAPT acceptera le signataire d'un paquet à partir du moment que :

- le certificat du paquet est également présent dans le répertoire *public\_certs\_dir*. Le test d'égalité est fait avec l'empreinte du certificat ;
- le certificat de l'Autorité ayant signé le certificat du paquet est présent dans le répertoire *public\_certs\_dir*. La recherche est faite avec l'attribut *issuer\_subject\_hash* du certificat. La signature du certificat est effectuée par la classe **x509.verification.CertificateVerificationContext** ;

### 77.8.8 Processus de signature d'un paquet

Le processus de signature du paquet est lancé lors des actions suivantes :

- action `wapt-get.exe build-upload <directory>`.
- action `wapt-get.exe sign-package <path-to-package-file.wapt>`.
- commande shell `wapt-signpackage.py <WAPT-package-list>`.
- sauvegarde d'un paquet *host* dans la console WAPT.
- sauvegarde d'un paquet *group* dans la console WAPT.
- import direct d'un paquet depuis un dépôt externe.
- wizard de création de paquets à partir de MSI et de setup.

#### Paramètres initiaux

- Fichier ZIP du paquet ;
- clé privée RSA du signataire encodée en format *.pem* et chiffrée (par l'algorithme *aes-256-cbc* de openssl si la clé a été créée dans la console WAPT) ;
- certificat *X509* du signataire correspondant à la clé privée ;
- si le paquet à signer contient un fichier `setup.py`, le certificat *X509* doit avoir l'extension *advanced Key Usage : codeSigning (1.3.6.1.5.5.7.3.3)* ;

## Signature des attributs du fichier control

Le fichier `control` d'un paquet décrit les métadonnées du paquet, en particulier son nom, sa version, ses dépendances et ses conflits. C'est la fiche d'identité du paquet.

Ces métadonnées sont primitivement utilisées par l'agent WAPT pour déterminer si un paquet doit être mis à jour, et quels autres paquets doivent être installés ou désinstallés préalablement.

Ces informations sont donc signées pour garantir aux Postes client leur intégrité et leur authenticité.

Étapes du processus :

- Les attributs `signed_attributes`, `signer`, `signature_date`, `signer_certificate` sont ajoutés à la structure du fichier `control` :
  - `signed_attributes` : liste des noms d'attributs avec séparateur virgule (,);
  - `signer` : `commonName` de l'objet du certificat du signataire;
  - `signature_date` : date et heure en cours (UTC) sous la forme “%Y-%m-%dT%H:%M:%S”;
  - `signer_fingerprint` : empreinte sha256 du certificat encodée en hexadécimal obtenue par la fonction `fingerprint` de la classe `cryptography.x509.Certificate`.
- Les attributs de la structure `control` sont encodés en JSON.
- Le JSON BLOB résultant est signé avec un hachage `sha256` et un remplissage `PKCS#1 v1.5`.
- La signature est encodée en base64 et stockée dans le JSON dans l'attribut `signature` du fichier `control`.

## Signature des fichiers du paquet

- Les attributs du fichier `control` sont signés et sérialisés en JSON. Le résultat est stocké dans le fichier `<WAPT>control` du ZIP du paquet.
- Le certificat X509 du signataire depuis le fichier est stocké dans le fichier `<WAPT>certificate.crt` du paquet WAPT.
- Les empreintes `sha256` de tous les fichiers contenus dans le paquet WAPT sont codées en hexadécimal et stockées sous forme de liste JSON [(nom de fichier, hachage),] dans le fichier `<WAPT>manifest.sha256` dans le paquet WAPT.
- Le contenu du fichier `<WAPT>manifest.sha256` est signé avec la clé privée de l'*Administrateur* (clé RAS 2048 bits), avec un hachage `sha256` et un remplissage `PKCS#1 v1.5` :
  - La procédure de signature fait appel à la fonction `sign` de la classe `cryptography.rsa.RSAPrivateKey.signer`.
  - `cryptography.rsa.RSAPrivateKey.signer` repose sur les fonctions OpenSSL de `EVP_DigestSignInit`.
- La signature est encodée en base64 et stockée dans le fichier `<WAPT>signature.sha256` du paquet WAPT.

### 77.8.9 Vérifier la signature des attributs d'un paquet

La vérification a lieu :

- Lors de la mise à jour de l'index des paquets disponibles sur le client WAPT à partir de l'index `Packages` du dépôt.
- Lorsqu'une signature de paquet est vérifiée (installation, téléchargement) lorsqu'elle n'est pas en mode *développement*, c'est-à-dire si l'installation se fait à partir d'un fichier ZIP et non d'un répertoire de développement.

La vérification consiste à :

- Lire les attributs du fichier `control` depuis le fichier `<WAPT>\control` du ZIP du paquet.
- Récupérer le certificat X509 du signataire depuis le fichier `<WAPT>\certificate.crt` du ZIP du paquet.
- Décoder l'attribut signature du `control` depuis le format base64.
- Construire une structure JSON avec les attributs devant être signés (tels que définis dans la classe `PackageEntry`).
- Vérifier si la clé publique du certificat du titulaire peut vérifier le hachage de la liste structurée des attributs JSON et la signature du fichier `control`, en utilisant le hachage `sha256` et le remplissage `PKCS#1 v1.5`.
- Vérifier si le certificat est de confiance (soit présent en tant que tel dans les certificats de confiance, soit signé par une *Autorité de Certification* de confiance).

Dans le cas où nous devons vérifier les attributs sans avoir le paquet WAPT à disposition, nous récupérons la liste des certificats des détenteurs potentiels de certificats à partir du fichier d'index `Packages` sur le dépôt WAPT. Les certificats sont nommés `ssl/<hexadecimal formatted certificate fingerprint>.crt`.

Un attribut de la structure `control` du paquet indique l’empreinte du certificat du signataire du fichier `control`.

### 77.8.10 Vérifier la signature d’un paquet

La vérification a lieu :

- Lors de l’installation d’un paquet sur un Poste client.
- Lors de l’édition d’un paquet existant.
- Lors de l’import d’un paquet depuis un dépôt externe (si option cochée dans la console).

La vérification consiste à :

- Récupérer le certificat X509 du signataire depuis le fichier `<WAPT>\certificate.crt` du ZIP du paquet.
- Vérifier que le certificat a été signé par une autorité de confiance dont le certificat est présent dans le fichier `ssl` du client WAPT.
- Vérifier la signature du fichier `<WAPT>\manifest.sha256` avec la clé publique.

### 77.8.11 Signature d’une action immédiate

Depuis la console, les *Administrateurs* peut déclencher des actions directes sur le client WAPT, s’il est connecté au serveur par le mode Websockets.

La console WAPT signe ces actions avec la clé et le certificat de l’*Administrateur* avant de les envoyer au serveur WAPT en utilisant une requête HTTPS POST ; la requête est ensuite transmise aux clients WAPT ciblés.

Les actions possibles sont :

- `trigger_host_update`.
- `trigger_host_upgrade`.
- `trigger_install_packages`.
- `trigger_remove_packages`.
- `trigger_forget_packages`.
- `trigger_cancel_all_tasks`.
- `trigger_host_register`.

### Processus de signature pour des actions immédiates

- L’action est définie par son nom et des attributs dépendants de l’action. Les attributs sont *uuid*, *action*, *force*, *notify\_server*, et *packages* (pour les actions impliquant une liste de paquets).
- Les attributs *signed\_attributes*, *signer*, *signature\_date*, *signer\_certificate* sont ajoutés à la structure de l’action :
  - *signed\_attributes* : liste des noms des attributs qui sont signés.
  - *signer* : `commonName` de l’objet du certificat du signataire.
  - *signature\_date* : date et heure en cours (UTC) sous la forme “%Y-%m-%dT%H:%M:%S”.
  - *signer\_certificate* : certificat X509 du signataire encodé en base64.
- La structure est encodée en JSON.
- La signature du JSON est calculée à partir de la clé privée RSA du signataire en utilisant un algorithme de hachage *sha256* et un remplissage *PKCS1 v1.5*.
- La signature est encodée en base64 et stockée dans le JSON dans l’attribut *signature*.



## Vérifier la signature d'une action immédiate

Depuis la console, les *Administrateurs* peut déclencher des actions directes sur le client WAPT, s'il est connecté au serveur par le mode Websockets.

Les actions sont encodées en JSON, signées avec la clé et le certificat de l'*Administrateur* et relayées vers le client WAPT visé par le serveur WAPT.

Les actions possibles sont :

- `trigger_host_update.`
- `trigger_host_upgrade.`
- `trigger_install_packages.`
- `trigger_remove_packages.`
- `trigger_forget_packages .`
- `trigger_cancel_all_tasks.`
- `trigger_host_register.`

L'action `get_tasks_status` ne demande pas d'authentification ssl.

Sur réception d'un évènement par la connexion Websocket du client WAPT :

- Le certificat X509 du signataire de l'action est extrait du json (format PEM).
- Le client WAPT teste si le certificat est un certificat de confiance, c'est-à-dire présent dans `<WAPT>ssl` ou signé par une autorité de confiance (certificat de l'autorité présent dans `<WAPT>ssl`).
- Le client WAPT teste si le certificat peut vérifier la signature présente dans la structure JSON de l'action ce qui consiste en :
  - Extraire la signature encodée en base64 dans le json depuis l'attribut *signature* dans le fichier JSON;
  - Extraire la date de signature formatée sous la forme “%Y-%m-%dT%H:%M:%S” depuis l'attribut *signature\_date*;
  - Vérifier que la date de signature n'est pas trop ancienne ou dans le futur de plus de 10 minutes;
  - Reconstruire une représentation json des attributs de l'action;
  - Vérifier que la clé publique du certificat peut vérifier le JSON avec la signature en utilisant un algorithme de hachage *sha256* et un remplissage *PKCS1 v1.5*.

### 77.8.12 Vérification du téléchargement complet d'un paquet

Pour chaque paquet, une somme *md5* du paquet est calculée et disponible dans l'index *Packages* du dépôt.

Lors de l'installation d'un paquet, le client vérifie si le paquet est déjà disponible localement dans le répertoire `<WAPT>\cache`.

Si le fichier est présent, sa somme *md5* est comparée avec la somme *md5* présente dans l'index. Si elles diffèrent, le paquet en cache local est effacé.

**Cette somme md5 ne sert qu'à s'assurer qu'un paquet a été téléchargé complètement.**

La vérification de la signature du paquet sera utilisé à la place de la somme *md5* et d'être effectivement assuré de l'intégrité et de l'authenticité du paquet.



---

## Appliquer les meilleures pratiques au packaging de logiciels

---

---

**Note :** `_benwa` est un administrateur système et il a autorisé Tranquil IT à republier son excellente diatribe sur reddit [Developers, you can make sysadmins happier](#).

---

### 78.1 Variables d'environnement

- Les variables d'environnement [existent depuis le DOS](#). Elles peuvent vous faciliter la vie (et la mienne).

### 78.2 Répertoires des programmes

- Tous les systèmes n'utilisent pas `C:\` comme lecteur principal. Certaines entreprises utilisent la redirection de dossiers, et déplacent le dossier Documents. Certains endroits dans le monde ne parlent pas anglais et leurs noms de répertoires reflètent cela. **Utilisez ces variables d'environnement pour que vos programmes fonctionnent tout simplement :**
  - `%SystemDrive%` est le lecteur où se trouve `%SystemRoot%`. Vous n'avez probablement pas besoin de le savoir ;
  - Le système d'exploitation Windows est situé dans le répertoire `%SystemRoot%`. Ne vous en souciez pas. Laissez le répertoire Windows tranquille ;
  - `%ProgramFiles%` est l'endroit où vous devez placer vos fichiers de programme, de préférence dans une structure `Company\Program` ;
  - `%ProgramFiles(x86)%` est l'endroit où vous devez placer vos fichiers de programme 32 bits. Veuillez les mettre à jour pour le 64 bits. Le 32-bit ne sera plus supporté dans l'avenir, et les entreprises attendront que vous vous organisiez pour bien plus longtemps que nécessaire ;
  - `ProgramData%` est l'endroit où vous devez stocker les données qui ne sont pas spécifiques à l'utilisateur, mais qui doivent quand même être écrites par les utilisateurs (les utilisateurs n'ont pas non plus d'accès en écriture à ce dossier). Votre programme ne devrait pas nécessiter de droits d'administrateur pour s'exécuter, car vous ne devriez pas nous faire écrire dans le répertoire `%ProgramFiles%`. Aussi, ne mettez pas d'exécutables dans ce répertoire.

- %Temp% est l'endroit où vous pouvez traiter des données temporaires. Placez ces données dans un nom de dossier unique (peut-être un GUID généré) afin de ne pas provoquer d'incompatibilité avec un autre programme. Windows fera même le nettoyage à votre place. Ne placez pas de données temporaires dans les dossiers %ProgramData% ou %ProgramFiles%;
- %AppData% vous permet de sauvegarder les paramètres de l'utilisateur qui exécute votre programme. C'est un endroit fantastique qui peut être synchronisé avec un serveur et être utilisé pour migrer rapidement et facilement un utilisateur vers une nouvelle machine et conserver tous les paramètres de ses programmes. Ne mettez pas de fichiers géants ou éphémères ici.  
Vous pourriez être à l'origine d'une connexion très lente si vous mettez les mauvais éléments ici et qu'une machine doit les synchroniser. **NE METTEZ PAS VOS FICHIERS DE PROGRAMMES ICI.** C'est l'entreprise qui décide quels logiciels sont autorisés à fonctionner, ce n'est pas à vous de décider, ni aux utilisateurs qui ne savent peut-être pas comment l'environnement de leur entreprise est configuré;
- LocalAppData% permet de placer des fichiers plus volumineux spécifiques à un utilisateur ou à un ordinateur. Par exemple, personne n'a besoin de synchroniser un cache de vignettes. Elles ne seront pas transférées lorsqu'un utilisateur migrera vers une nouvelle machine, ou se connectera à une nouvelle station VDI, ou à un nouveau serveur de terminal. **NE METTEZ PAS VOS FICHIERS DE PROGRAMME ICI NON PLUS;**

---

**Note :** De plus en plus d'éditeurs de logiciels proposent des versions *portables* de leurs logiciels qui s'installent et s'exécutent à partir de %AppData% ou de %LocalAppData%. Votre objectif est de permettre aux utilisateurs d'installer des logiciels même s'ils ne sont pas Administrateurs Locaux et vous commercialisez cela comme une fonctionnalité, bien qu'il s'agisse plutôt d'un NOGO de sécurité. Pire encore, vous avez tendance à rendre difficile de trouver le bon *MSI* qui permettrait à vos clients d'installer correctement votre logiciel dans %ProgramFiles%. Faites en sorte qu'il soit facile de trouver votre *MSI* qui s'installera dans les %ProgramFiles%, de cette façon vous ferez en sorte que les politiques de restriction des logiciels et de verrouillage des applications de vos clients fonctionnent bien et que leurs administrateurs système soient satisfaits.

Vous pouvez aussi bien obtenir ces chemins de répertoires par des appels [API](#) si vous n'utilisez pas ou ne pouvez pas utiliser de variables d'environnement.

---

## 78.3 Logs

- Utilisez le [Journal des événements Windows](#) pour la journalisation. Il s'occupera de la rotation pour vous et un administrateur système peut transmettre ces journaux ou faire ce qu'il faut. Vous pouvez même créer votre propre petite zone juste pour votre programme.

## 78.4 Codes d'erreur

- Utilisez les [codes d'erreur standard](#) lorsque vous quittez votre programme.

## 78.5 Impression

- Utilisez l'[API d'impression Windows](#) et n'utilisez pas l'impression directe dans votre programme.

## 78.6 Distribution

- Distribuez votre programme en MSI. C'est le standard pour les fichiers d'installation de Windows (même si Microsoft ne l'utilise pas toujours lui-même).
- [Signez vos fichiers d'installation et vos exécutables](#). C'est ainsi que nous savons que votre MSI est valide et que nous pouvons le mettre sur une liste blanche dans AppLocker ou équivalent.

---

**Note :** Applocker et [Software Restriction Policies](#) peuvent être très efficaces et la **gestion de ces stratégies peut être rendue plus simple avec WAPT**.

---

## 78.7 Mises à jour

- Vous souhaitez que votre programme se mette à jour ? C'est possible si l'entreprise est d'accord. Vous pouvez créer une tâche ou un service programmé qui s'exécute en mode élevé pour permettre cela sans accorder de droits d'administrateur à l'utilisateur. J'aime la façon dont Chrome Enterprise le fait : il donne une GPO pour définir les paramètres de mise à jour, la version maximale à laquelle elle va se mettre à jour (disons 81.\* pour permettre toutes les mises à jour mineures automatiquement et les versions majeures sont manuelles), et un service. Ils ont également une GPO pour empêcher les installations lancées par les utilisateurs ;

---

**Note :** WAPT est conçu pour les entreprises qui ne permettent pas à leurs utilisateurs d'exécuter des mises à jour logicielles, c'est la politique souvent choisie par les grandes entreprises consciencieuses vis à vis de la sécurité.

---

## 78.8 Numéros de version

- Utilisez le [versionnage sémantique](#) (doit aller dans la propriété de version dans le fichier d'installation et dans la liste Ajout/Suppression de programmes, pas dans le titre de l'application) et ayez un [changelog](#). Vous pouvez également mettre à disposition en téléchargement votre installateur à un endroit prévisible pour permettre l'automatisation. Un chemin de mise à jour publié est également utile ;

---

**Note :** Si vous appliquez cette pratique, alors vous rendrez les administrateurs système qui déploient vos mises à jour logicielles en utilisant la fonction `WAPT def_update()` **très heureux !**

---

## 78.9 GPO

- Les modèles ADMX sont des trucs très moches ;

---

**Note :** Nous sommes tout à fait d'accord avec vous [\\_benwa](#) sur ce point chez Tranquil IT. Si les développeurs conseillent à leurs clients d'utiliser des GPO pour déployer leur logiciel ou leur système ou les paramètres des utilisateurs, alors, **ils doivent apprendre que les GPO ne sont pas fiables**.

Au lieu de cela, packagez vos logiciels, votre système et vos configurations utilisateur en utilisant WAPT. Un fichier `setup.py` est beaucoup plus facile qu'un fichier `xml` pour les administrateurs système qui doivent le vérifier avant de le déployer.

Les paquets WAPT peuvent être appliqués récursivement à des arbres d'Unités Organisationnelles, de sorte que votre paquet WAPT se comportera en production exactement comme le ferait une GPO, **juste beaucoup plus facilement**.

---

## 78.10 Dongles de licences

- Les dongles de licence USB sont un péché. Utilisez une licence de logiciel ordinaire ou une licence activée par réseau. Je suis sûr qu'il y a plein de systèmes de gestion des licences sur le marché pour que vous n'ayez pas à réinventer la roue ;

---

**Note :** Vous pouvez faire en sorte que votre logiciel accepte une clé de licence comme paramètre dans votre exécutable *msi*.

WAPT peut être utilisé pour attribuer des clés de licence à des postes de travail individuels lors de l'installation en utilisant une méthode *qui garantit que la clé de licence ne peut pas être lue pendant le transport*.

Ensuite, si vous voulez que votre logiciel appelle chez vous pour vérifier la validité de la licence, faites en sorte que votre méthode fonctionne avec des *serveur mandataires*.

---

## 78.11 Fonctionnement en réseau

- N'utilisez pas ce fichu champ de saisie IPv4 personnalisé. Utilisez des FDQN. L'IPv6 existe depuis 1998 et fonctionnera avec votre logiciel si vous lui donnez une chance ;
- Le pare-feu Windows (je ne peux pas vraiment en dire plus sur les pare-feu tiers) va rester en place. Connaissez la différence entre une règle entrante et une règle sortante. Il est probable que votre serveur aura besoin d'une règle entrante. Vos clients n'auront probablement même pas besoin d'une règle sortante. Configurez-les au moment de l'installation, et non au moment du lancement. Utilisez les groupes de pare-feu pour faciliter le filtrage. N'utilisez aucune règle si vous pouvez. Le but n'est pas de faire fonctionner le système, mais de le faire fonctionner en toute sécurité. Si vous n'utilisez pas les numéros de version dans vos chemins d'installation, vous n'aurez peut-être même pas à refaire ces règles après chaque mise à jour ;
- Les serveurs mandataires sont bons pour l'hygiène et les serveurs mandataires sont maintenant une caractéristique de sécurité par défaut non seulement dans les environnements informatiques des entreprises, mais aussi sur les petits réseaux. En rendant votre logiciel non compatible avec les proxies, les administrateurs réseau de vos clients devront établir et maintenir des règles spéciales pour leurs pare-feu, et cela rien que pour vos beaux yeux. Il est facile de coder votre logiciel pour qu'il fonctionne avec des serveurs mandataires, alors faites-le !

## 78.12 PDFs

- Ne livrez pas un logiciel qui nécessite d'autoriser le fonctionnement de javascript dans les lecteurs de PDF. La logique métier doit être exécutée avant la sortie au format PDF, pas après.

---

**Note :** Le *PDF* est le format de fichier que les gens utilisent par défaut pour échanger des documents. Les lecteurs PDF sont destinés à afficher des documents, et non à exécuter des programmes non signés.

---

---

## Stratégie de sortie des mises à jour de WAPT

---

### Les mises à jour WAPT ne sortent pas selon un calendrier fixe.

Au lieu de cela, Tranquil IT sortira une nouvelle version majeure de WAPT lorsque de nouvelles mises à jour fonctionnelles majeures seront intégrées au cœur du produit.

Tranquil IT publiera des versions mineures intermédiaires de WAPT entre les versions majeures afin de corriger les défauts de fonctionnement et de sécurité.

### 79.1 Délai de publication entre les versions Enterprise et Discovery

Une nouvelle version majeure sera disponible en une RC1 (Release Candidate #1) comme **Enterprise** et cette même version sortira en **Discovery**. Avant de la sortir, la version Enterprise aura fait l'objet de tests internes approfondis avec des clients ayant l'**Insider Program** pour s'assurer qu'aucune régression ne s'imisce dans le cœur de WAPT.

La version Enterprise passera par plusieurs RCs et la version finale de disponibilité générale Enterprise sera alors disponible entre 4 et 8 semaines après la sortie de la première version de nos clients avec un **Program Insider**.

Ce délai apportera plusieurs bénéfices au processus de sortie de la version Enterprise :

- Accorder plus de temps pour effectuer des tests approfondis des nouvelles fonctionnalités Enterprise tout en évitant les régressions.
- Permettre à Tranquil IT de travailler avec un petit groupe de clients Enterprise sélectionnés afin d'assurer le bon déroulement des procédures de mise à niveau.
- Donner un peu de temps au forum et à la liste de diffusion pour indexer les questions et les réponses qui seront éventuellement incluses dans la documentation officielle.
- Permettre à l'équipe de documentation de s'appuyer sur une base fonctionnelle figée pour ainsi documenter de manière fiable les fonctionnalités nouvelles ou améliorées.
- Donner à l'équipe de traduction le délai nécessaire pour mettre à jour les traductions.
- Permettre à l'équipe de communication et marketing de s'appuyer sur une base fonctionnelle figée pour ainsi rétro-planifier les annonces, les podcasts vidéo et la promotion générale de WAPT.





### 80.1 WAPT-2021-01 : CVE-2021-38608

- Brief : Insecure permission allows a user running as guest to escalate privileges.
- Announced : August 13, 2021.
- Impact : **High**.
- Products : WAPT Enterprise & Community.
- Impacted versions : WAPT Enterprise < 2.0.0.9450, WAPT Enterprise < 1.8.2.7373 and WAPT Community < 1.8.2.7373.
- Description : Insecure permission allows guest OS users to escalate privileges via WAPT Agent.
- Reporter : Anass ANNOUR from the ORM/ITT&AC Risk Assessment Team, BNPParibas.
- Published CVE : <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-38608>.



### 81.1 WAPT-2.1 Serie

#### 81.1.1 WAPT-2.1.2.10678- (2022-12-01)

hash : 7f49ee61

This is a security release with a few related bugfixes. All Wapt 2.0 version below 2.2.3.12481 are affected.

Note : if you are using WAPTAgent deployment though GPO, don't forget to update your waptdeploy binary in the definition GPO.

#### WAPT Core

- [SEC] upgrade python from 3.8.13 to 3.8.15
- [SEC] upgrade openssl from 1.1.1k to 1.1.1s
- [SEC] upgrade agent kerberos lib from 1.19.3 to 1.20.1 (linux/mac)
- [SEC] upgrade python modules with CVEs
  - pylint==2.12.2 -> 2.15.6
  - ujson==4.0.2 -> 5.5.0
  - waitress==2.0.0 -> 2.1.2

### WAPT Agent

- [SEC] waptdeploy.exe. Use only wapt\_is1 install location from registry to get the current wapt install dir. don't run wapt-get to check working condition.

### 81.1.2 WAPT-2.1.2.10652 (2022-01-10)

hash : 7dd63b61

- [UPD] shorten the default package filename. If target\_os is alnum, don't include md5sum of in the filename If target\_os is in tags, don't duplicate it in filename
- [FIX] disable debug data for linux
- [FIX] try to circumvent issue with Trend antivirus blocking the WaptTaskManager . Looks like the issue is with platform.win32\_ver using win32api.GetVersionEx...
- [FIX] Installed softwares invalid conditions
- [FIX] fix local\_user and local\_group macos
- [FIX] removed workaround on 60s delay for websocket disconnect
- [FIX] use CompressGZip instead of CompressZLib on waptserver, compression is GZip
- [FIX] Allow "~" in package filenames
- [FIX] try to not update records in database if data has not changed
- [FIX] Wake on lan relay now equals is remote repository, close #2940
- [FIX] fix group members
- [FIX] return only local and user group (ignore nsswitch)
- [FIX] backport waptexit (improved detailed logging) from 2.2
- [FIX] backport waptlicences py module from 2.2
- [SEC] check that hostname matches https cert in waptconsole http client.
- [FIX] backport uwaptlicencing : allow empty json licencing data
- [FIX] fix WaptHttpPostData
- [FIX] check valid uri in wapthttputils waptwget WaptWget\_Try
- [FIX] init LastModifiedDate to "" if not found in THttpResponse
- [FIX] add a 50ms report delay for httpprogressnotification
- isolate wapt python engine : PyFlags := [pfNoUserSiteDirectory, pfIsolatedFlag];
- [FIX] setuphelpers : backport changes from 2.2 is\_linux64 type\_rhel fix installed\_softwares for type\_redhat upd uninstall\_apt with autoremove
- [FIX] user\_appdata = user\_local\_appdata for unix
- [IMP] introduced get\_powershell\_str, get\_default\_app remove\_appx
- [IMP] introduce InitLogger for waptexit
- [FIX] waptconsole : generalize the use of a fallback package\_uuid in case of old packages without package\_uuid field.
- [FIX] waptconsole : use editable dropdown in frmpackagedetails for maturity
- [FIX] backport issue with inc version of some group packages when importing
- [FIX] Disable client side ssl authentication on root server url (regression)
- [FIX] isolate from user python env when building binary packages
- [UPD] improved feedback message for license activation on server.
- [UPD] wapt-scanpackages.py : add option -d to disable update of database Packages table.
- [FIX] The -b switch is True by default, so there were no way to disable update of database table.
- [UPD] waptconsole : be tolerant for old package without package\_uuid
- [UPD] strip ending slash in {{data.wapt.hostname}} server template properties to avoid double slashes in templates result
- [UPD] backport openssl build parameter from 2.2
- [FIX] waptagent url link in server index page.
- [FIX] setproctitle only for unix
- [FIX] locate packages in host packages grid using package\_uuid instead of id, so that refreshing grid works properly with a multiselection of hosts.

- [UPG][SEC] upgrade python version from 3.8.11 to 3.8.12
- [FIX] remove python3 dependencie. Now python3 is included in wapt

### 81.1.3 WAPT-2.1.2.10605 (2021-11-30)

hash : e2a0e2a0

- [FIX] waptconsole : backport edit multiple hosts add/remove depends/conflicts (issue « no password available yet » when kerberos enabled) backport IpExecute from 2.2
- [FIX] unable to edit stripped down package with integrated package editor. (setup.py file hash issue) update package size
- [FIX] bad path for nginx dhparam for windows server
- [FIX] upgrade mormot2
- [FIX] waptself local admin NOPASSWORD setting didn't work anymore log auth user when task is triggered from local wapt webservice don't raise exception in check\_auth\_groups but return (None, None) instead to avoid Error 500 in browser backport fix for integer attributes in packages index backport fix for loading ssl libraries
- [FIX] Update wake on lan with broadcasts
- [FIX] Error « Add : Unexpected [%] object property in an array » for old package with empty package uuid
- [FIX] Acl handle boolean as global ACL
- [FIX][SEC] issue with acls : action is enabled when acl is set to json false

### 81.1.4 WAPT-2.1.2.10588-rc1 (2021-11-22)

hash : e70d9039

- [FIX] fix installed\_softwares for older debian and improve inventory performance
- [FIX] fix glpi inventory failure (exception on int conversion)
- [SEC] [FIX] invalid condition on package hash check
- [SEC] [FIX] cleanup nginx config templates
- [NEW] add uwsgi support for Debian server
- [FIX] add user information in audit
- [FIX] Improve lazarus ini parser to support other values than "1"/"0" as boolean values (True, true, 1, 01, etc. same behavior as python iniparse)
- [IMP] support for message previsualisation and templates in waptmessage editor and better multiline support
- [UPD] waptsetup : don't use kerberos by default
- [NEW] show certificate when double click in acl tab
- [IMP] Don't propose to start waptconsole after install (due to different user context)

### 81.1.5 WAPT-2.1.1.10568 (2021-11-08)

hash : 978c00ae

This is a bugfix version with some small improvements. The main fix is for websocket issue.

- [IMP] prevent multiple websockets connections from same host uuid on server (bugged wapt clients can maintain multiple websockets, which leads to a lack of available connections on server)
- [FIX] fix restart waptservice with exit code 10 (managed by the nssm service manager)
- [FIX] fix case on waptservice where different threads access simultaneously to a shared Wapt instance
- [IMP] introduce some randomness when waptservice reconnects its websocket.
- [IMP] check more cases to determine if token for websocket has to be updated.
- [IMP] wait socket client until it is actually disconnected before trying to reconnect to avoid multiple websocket threads from same client.
- [IMP] don't recreate a new SocketIOClient at each reconnection, but reuse existing one to minimize risk of multiple connections.

- [FIX] don't consider “%” char as unsafe in filenames
- [IMP] improved logging of waptservice (logger wapttasks report main actions triggered by the service in waptlogwaptservice.log). Removed “flask.app” logger config.
- [IMP] remove packages's persistent directory on client when a package is forgotten
- [IMP] added ignore\_empty\_names arg to setuphelpers.installed\_softwares
- [IMP] displays package\_uuid with command wapt-get list
- [IMP] add “redhat\_based” tag for package os tags
- [FIX] fix decrypt\_fernet/fernet\_encrypt functions
- [IMP] report key as name in softwares inventory for softwares without a descriptive name
- [FIX] update properly the server\_uuid column in hosts database
- [FIX] fix removal of packages when “only\_if\_not\_process\_running” argument is True

Known issues :

- When the websocket is reconnecting, if the IP address has changed, the main IP address is not updated in IP address column in WAPT Console.

### 81.1.6 WAPT-2.1.0.10550 (2021-10-08)

hash : 953c9552

This is a bugfix version with some small improvements.

- [FIX] mass add / remove on multiple machine at once
- [FIX] issue when editing a package without a « description\_en » attribute in control file
- [FIX] fix drag drop when editing selfservice package
- [IMP] improve feedback when uploading packages
- [IMP] better handling of the list of wakeonlan relay
- [IMP] remote repository is now by default a wakeonlan relay
- [FIX] fix access violation error when viewing certificate list
- [FIX] do not enable verbose logging by default on waptconsole, waptexit and waptselfservice (might fill up %APPDATA% ...)
- [FIX] use templates/wapt-logo.png in waptexit if it exists
- [IMP] improve login error message

### 81.1.7 WAPT-2.1.0.10517 (2021-09-30)

hash : fa2af298

This is the first release of the 2.1 branch. It is mainly a incremental improvement with many small but worthy fixes on the 2.0 branch.

waptservice

- [IMP] during upgrade, wapt-get session\_setup is not run if no userspace config is define for the installed packages

waptdeploy

- [IMP] automatic proxy detection and configuration possible with the new `-http_proxy=0,1` parameter or explicit url command line parameter
- [IMP] disable https verification when downloading waptagent.exe if a fingerprint is provided (allows installation with on out-of-date computer with expired cert store)
- [IMP] do nothing if no `-waptsetupurl` argument is provided (it reduces the probability of false positive on antivirus check)
- [IMP] double check Wapt installed version after install and report error message if it does not match (allow detection of installation that have been blocked by a misconfigured antivirus for example)

waptconsole

- [NEW] tech preview : new tab to provide basic package editing fonctionnality directly in the waptconsole without having to open Pyscripter or VSCode

- [NEW] tech preview : new tab to browse the development directory directly from the console
  - [NEW] Single Sign On with Kerberos authentication (if `service_auth_type = waptserver-ldap` and `use_kerberos=1`)
  - [NEW] new button to display packages that have a specific package as a dependency in the private repo tab
  - [NEW] new message box to decrypt message sent by the agents (using `encrypted_data_str/print_encrypted_data` in `wapt-crypto`). This allows an admin to upload sensitive information from desktop that will be asymmetrically signed by the admin's public key
  - [NEW] new set of icons and many small visual improvements
  - [NEW] new software inventory tab to display installed software (not packages) and see which machines have that specific software.
  - [NEW] button to delete Windows Update KB files that are not used anymore by any computers. This allows to keep the Windows Update storage volume under control
  - [NEW] new tab to have a user-friendly display of the certificates that are deployed on a specific machine
  - [NEW] new tab to display the certificates that are available on a repository
  - [NEW] new warning icons on the machines tab when the computer needs a restart (after a windows update for example)
  - [NEW] new filter by OS option
  - [NEW] new icons in the OU tree view if a OU package exists for that Organizational Unit
  - [NEW] information message about the choice of maturity when creating new agent and by default uploading in DEV maturity (to avoid being directly deployed to all client computers, this allow to test the new Wapt agent on a subset of computer before full scale deployment)
  - [IMP] make GLPI export configuration more intuitive
  - [IMP] improve console plugin versatility. All inventory attribute can now be used in command lines (it use the « mustache » template syntax, eg. `{{ main_ip }} {{ computer_fqdn }} {{ host_capabilities.os_version }}` « `{{#host_capabilities.tags}}{.},{/host_capabilities.tags}` » etc
  - [IMP] allow non standard port in `waptconsole` configuration
- `waptself`
- [NEW] allow custom logo in `waptselfservice`
  - [NEW] Single Sign On using Kerberos (needs `service_auth_type = waptserver-ldap` and `use_kerberos=1`)
  - [IMP] allow customisation of package details view using template engine
- `waptexit`
- [IMP] allow custom logo (on Windows, Linux and macOS)
- `wapt-get`
- [NEW] better handling of licence information. Now the licence is uploaded on the server and it is not necessary to install it on every admin console computer
  - [IMP] propagate `ExitCode` from Python calls for better error handling
  - [IMP]
  - [IMP] better handling of websocket reconnection (check of socket status every 120s)
  - [IMP] periodic check of the UUID and the current certificate of the agent for consistency between the agent and the client computer
  - [NEW] `waptsetup` et `waptserversetup` new parameters : `set_verify_cert` and `set_kerberos`

## 81.2 WAPT-2.0 Serie

### 81.2.1 WAPT-2.0.0.9470 (2021-10-07)

hash : 5065cb57

This is a security release with a few related bugfixes. All Wapt 2.0 version below 2.0.0.9467 are affected

- [SEC] fix for vuln in `urllib3` CVE-2021-33503 (CVSS Score : 7.5 High, CVSS :3.1/AV :N/AC :L/PR :N/UI :N/S :U/C :N/I :N/A :H)

- [SEC] Sanitize filename used when downloading files on local client. (CVSS Score : 7.5 High, CVSS;3.1/AV :L/AC :H/PR :H/UI :N/S :C/C :H/I :H/A :H/E :U/RL :O/RC :C) Enforced on wget and local filenames for downloaded packages (chars “\” “.” @ | ( ) : / , [ ] < > \* ? ; ` \n are removed or replaced)
- [SEC] don't use PackageEntry filename attribute to build target package filename as it is not signed.
- [UPD] Wapt.remove : reraise exception if there is exception in uninstall script return traceback in “errors” key return code 3 if there are errors when removing packages in wapt-get remove
- [FIX] handles wildcards in certificates in waptconsole config and create waptsetup update UI in external repositories config when setting CA bundle
- [FIX] use PackageEntry.localpath only for local status of a package.
- [UPD] split PackageEntry non\_control\_attributes into repo\_attributes and local\_attributes local\_attributes are not put into Packages index as they are not relevant for remote access.
- [UPD] update python modules requirements following urllib3 upgrade idna==3.2 (from 2.10) certifi==2021.5.30 (from 2020.12.5) requests==2.26.0 (from 2.25) urllib3==1.26.6 (from 1.26.5)

### 81.2.2 WAPT-2.0.0.9450 (2021-08-10)

hash : 7bc6920c

This is a security fix version affected by [CVE-2021-38608](#).

Please visit the *security bulletin* to learn more.

### 81.2.3 WAPT-2.0.0.9449 (2021-06-22)

hash : 70283a14

This is a bugfix version with some small improvements.

WAPTAgent :

- [FIX] Windows Update fix in the progress bar
- [IMP] Allow WAPTAgent to upgrade even when on batteries

WAPTServer :

- [IMP] many fixes in GLPI sync
- [FIX] better handling of service\_delete exception cases
- [FIX] db migration handling with create\_defaults\_users procedure
- [FIX] on windows skip agent build if there is no available certificate for signing

Core :

- [IMP] improve compatibility of Packages file for easing upgrade from WAPT 1.8.2
- [IMP] WAPTDeploy improve behavior to avoid wrong red flag from AV softwares

### Caveat

For macOS support one should use WAPTAgent 2.1 version available in nightly channel.



### 81.2.4 WAPT-2.0.0.9428 (2021-05-06)

hash : 4b33cf96

This is a bugfix version with many small improvements.

Console :

- [IMP] Improve CreateWaptSetup form layout
- [IMP] restore focused column visibility when refreshing grid data
- [FIX] Fix wrong path for wapt-get.py in vscode project
- [UPD] Update No fallback in rules to true by default
- [FIX] enable-check-certificate with wildcard
- [FIX] take into account the use\_http\_proxy\_for\_repo ini setting (if not present, assume False)
- [FIX] fix setup\_package\_template\_msu.py.templ for package Wizard
- [IMP] Add new template for creating package with certificate
- [IMP] Add option to check downloaded package with VirusTotal in package import GUI
- [IMP] Add update-package source action directly in Private repository in waptconsole

Agent :

- [IMP] use task queue for the forced installs instead of running them inline
- [FIX] Database not opened when we check Hosts who are secondary repositories
- [IMP] restart partial download of Windows Update files
- [IMP] improve icons handling in WaptSelfService
- [IMP] on macOS use machine certificate store by default for https certificate validation
- [IMP] reload\_config\_if\_updated now reload config if public\_certs\_dir have changed
- [FIX] WUA : better handling of return code « does not apply to this computer »

Server :

- [FIX] Fix bad migration of PGSQL DB server side
- [FIX] improve db upgrade in corner cases

Setuphelpers :

- [FIX] register\_windows\_uninstall calculation and using correct x86\_64 environment with register\_uninstall and unregister\_uninstall
- [IMP] improve inline function description for documentation

### 81.2.5 WAPT-2.0.0.9343 (2021-04-08)

hash : 117d62b8

This is mainly a bugfix release after the initial 2.0.0 release.

Console :

- [IMP] show an explicit message if user can't build a customized waptagent
- [IMP] enable remote repo sync if there are repo configured (making remove\_repo\_support parameter obsolete)
- [IMP] better filtering on maturities
- [FIX] Fix for templates of vscode

Server :

- [IMP] include certificates from WaptUsers table in result of /api/v3/known\_signers\_certificates

WAPT ACL handling :

- [UPD] acl : Add an action to show the user certificate
- [UPD] creates default (empty) WaptUserAcls record on user login even for non ldap logins
- [IMP] better naming for acl domains

Setuphelpers :

- [FIX] Fix register\_uninstall
- [FIX] don't change silently maturity and locale in check\_package\_attributes

- [FIX] regression in wget resume

Other technical stuff :

- [IMP] add support for installation on OracleLinux
- [FIX] tighten files ACLs on Linux + fixes + SELinux fixes in postconf
- [IMP] introduce mORMot2 framework in Lazarus code
- [FIX] datetime conversion in console

### 81.2.6 WAPT-2.0.0.9300 (2021-03-30)

hash : 018b8b57

This is the first release of the 2.0 series. After one year in development and more than 1600 commits it brings a bunch of new features and enhancement to the last major update of WAPT 1.8.2. On the technical side WAPT 2.0 now embed Python3 and now support 8 new platforms (some of them backported to 1.8.2).

The switch to Python3 may require minor adjustment to the existing package that may have been development in-house (refer to the corresponding doc page). The packages offered by Tranquil IT through the WAPT Store are already compatible with WAPT 2.0.

#### From a sysadmin point of view

- ACLs
- [IMP] server side ACLs in addition to certificate validation.
- [IMP] user management interface with certificate listing.
- WaptConsole :
- [IMP] gui : change maturity directly from console.
- [IMP] gui : all WAPT package types are grouped in one tab.
- [IMP] helpers : build and upload locally development package from the console.
- [IMP] helpers : import default reporting queries from internet.
- [IMP] helpers : restart waptagent and restart client computer from console.
- [IMP] package wizard : support for RPM/DEB/PKG/DMG.
- [IMP] remote repositories : status bar for progression of creation/update of file `:sync.json` for repo sync.
- [IMP] windows update : new search bar, view machine with specific KB.
- [IMP] faster import and resigning of package, change of maturity, etc.
- [IMP] waptmessage : better handling of user oriented notification.
- [IMP] better logging of console actions and agent activity.
- Performance improvements for larger installation :
- [IMP] better handling of insert / update of inventory.
- [IMP] better handling of websocket updates.
- [IMP] GLPI integration : synchronize WAPT inventory to GLPI server.
- Better OS integration :
- [IMP] TLS certificate handling : **certifi** use local OS cert store instead of Python **certifi** integrated certstore.
- [IMP] increase the number of supported platform, improve packaging for Linux (deb and rpm) with support for arm64 and macos BigSur 64bit waptagent.
- Package development :
- [IMP] improved package wizard.
- [IMP] many small fixes and improvements to setuphelpers and better support for Linux and MacOS.
- [IMP] improve os targeting : now you can specify targeted OS and specific version of OS : eg. Debian(>=9,<=10).

## From a technical point of view

- Python : switch from Python2.7 to Python3 :
- Linux : use of venv by default with distrib python 3 version.
- Windows : switch python3 install to embedded edition 3.8.7.
- Different installer for WinXP / WinVista / Win2k3r2 / win2k8 (nonr2) (recent CPython version does not support older Windows systems anymore).
- Better handling of passwords with special chars.
- Upgrade WAPT core libs and scripting environment :
- upgrade to Python3 and Python libraries, change of kerberos and websocket libraries.
- upgrade of lazarus 3.0.10 and FPC 3.2.

## Caveat

- Support for non supported Windows version (WinXP, WinVista, Win2k8 (non-R2) and Win2k3) is still baking in the oven and should be ready shortly after the 2.0 release date.
- Redhat8 / CentOS8 : for upgrade it is necessary to remove WAPT selinux rules before using postconf again

## 81.3 WAPT 1.8 Serie

### 81.3.1 WAPT-1.8.2.7393 (2021-11-16)

hash : 75a5de09

This is a security release. All Wapt 1.8 version below 1.8.2.7393 are vulnerable.

- [SEC] upgrade babel python module from 2.5.1 to 2.9.1
- [UPD] **python lib upgrades urllib3, and requests**  
     chardet==4.0.0 requests==2.26.0 urllib3==1.26.7

### 81.3.2 WAPT-1.8.2.7388 (2021-10-07)

This is a security release. All Wapt 1.8 version below 1.8.2.7388 are vulnerable.

Security changelog wapt-1.8.2.7388\*

- [SEC] fix for vuln in urllib3 CVE-2021-33503 (CVSS Score : 7.5 High, CVSS :3.1/AV :N/AC :L/PR :N/UI :N/S :U/C :N/I :N/A :H)
- [SEC] Sanitize filename used when downloading files on local client. (CVSS Score : 7.5 High, CVSS ;3.1/AV :L/AC :H/PR :H/UI :N/S :C/C :H/I :H/A :H/E :U/RL :O/RC :C) Enforced on wget and local filenames for downloaded packages (chars “\” “.” @ | ( ) : / , [ ] < > \* ? ; ` `n are removed or replaced)
- [SEC] don't use PackageEntry filename attribute to build target package filename as it is not signed.
- [FIX] Waptconsole config : When retrieving server side https certificate don't write UTF16 string for in waptconfig. Remove wildcards from CN of certificate to compose cert filename.
- [UPD] **update python modules requirements following urllib3 upgrade**  
     certifi==2021.5.30 chardet==3.0.2 idna==2.8 requests==2.21.0 urllib3==1.24.3

### 81.3.3 WAPT-1.8.2.7373 (2021-08-10)

hash : e96e569c

This is a security fix version affected by [CVE-2021-38608](#).

Please visit the *security bulletin* to learn more.

### 81.3.4 WAPT-1.8.2.7372 (2021-06-21)

WAPTAgent :

- [FIX] fix regression on macos build after dependency upgrade
- [FIX] `_update_db` : error in for the calculation of `next_update_on` forpackage attributes `valid_until` and `forced_install_on`
- [IMP] be sure to not use `waptguihelper` when running as system user
- [UPD] add `-use-gui` for `vscode` / `pyscripter` build-upload of package

WAPTServer :

- [FIX] Fix regression on proxy setting for `waptserver`

Setuphelpers :

- [IMP] add `func split_arg_string` to split a command line into executable / args list

### 81.3.5 WAPT-1.8.2.7357 (2021-02-09)

WaptCore :

- [FIX] Be tolerant with `target_os = "all"` in windows.
- [FIX] in `installed_softwares`, ignore error when key can not be opened because of encoding issues (`_winreg` does not handle unicode, but ansi).
- [IMP] shown "" instead of None in `wapt-get` tables.
- [FIX] Update timestamping server and openssl hash : <http://timestamp.globalsign.com/scripts/timestamp.dll>.
- [FIX] be tolerant if no "id" attribute in installed packages report.
- [FIX] match properly packages with `target_os = all`.
- [IMP] prepare `installed_packages` for upgrade to wapt 1.9.
- [IMP] add `Timeit` class for test purposes.
- [IMP] disable sending unused data `waptwua_rules_packages`.
- [FIX] `waptupgrade` regression Bug introduced in Revision : 85686e4d631adb6e13b25146f3a81f3c09ca082d.
- [FIX] CA certificate PEM string stored as utf16 in certificate chain when creating a certificate signed by a CA (enterprise).

WaptConsole :

- [IMP] increase width of AD-Site combobox.
- [IMP] report packages `install_id` in console.

WaptAgent :

- [IMP] `wapt-get` : add `-newest-only` for search.
- [IMP] `waptexit` : add `ExceptionHandler`. Change the way exceptions are handled in threads to try to fix issues when `waptexit` hangs and can not be closed.

WaptServer :

- [FIX] don't actually update the listening websocket session ID if it is already set in hosts table.
- [IMP] `wapttasks` : force remove tasks locks at service startup.

### 81.3.6 WAPT-1.8.2.7334 (2020-12-03)

hash : 2d15afd9

This is a bugfix release. Ubuntu 16.0.4 amd64 and Debian 10 armhf clients are now supported.

#### Fixes and enhancements

- [FIX] fix base proxy string « » when editing a profile package.
- [FIX] fix « Unable to create file » when editing a profile package.
- [FIX] don't allow to save a self service rules packages without a name.
- [FIX] fix Access violation when importing from file.
- [FIX] fix issue with download\_icons.
- [FIX] improve search in waptconsole (search on concatenation of software name and software version).
- [FIX] fix extract CN from ssl client cert authentication for get\_auth\_token when windows client computer has an organization (in this case client csr/cert has a CN=<uuid>,O=<org> subject).
- [FIX] fix regression on wakeonlan introduced by backported code from 1.9.
- [FIX] PostgreSQL DB not correctly migrating from some 1.8.1.X.
- [IMP] Add key param for install\_msi\_if\_needed in setuphelpers\_windows.py.
- [FIX] Fix for no\_fallback in repositories rules.
- [FIX] Soupsieve python lib is set to 1.9.6 in requirements because later version are Python3 only.
- [FIX] Patch for SocketIO with proxy.
- [FIX] Fix triggers for repository sync in PostgreSQL who were not correctly migrated (Enterprise only).
- [IMP] Two new builders for both server and agent : Ubuntu 16.0.4 LTS / ARM x86 Debian 10 (Enterprise only).
- [IMP] Revert dhparam bits size to 1024 bits in Windows WAPT Server because it took too much time to generate. It can be generated afterward.
- [IMP] Increase default clockskew for signed action to 6 hours (before it was only 1 hour).
- [FIX] waptcrypto : security fix : prevent infinite loop in SSLCABundle.certificate\_chain if issuer cert and signed cert have the same subject but one has no authority\_key\_identifier.
- [FIX] waptcrypto : fix revoke\_cert, handle list of DNS names for certificates, fix AuthorityKeyIdentifier when neregating certificate from CSR.
- [FIX] waptservice fix for verify\_cert\_ldap in waptagent.
- [FIX] Patch pltis\_utils to display properly long integer in WAPTWUA. The wsusscn2.cab file may report KBs with incorrect huge download size up to 1TB.
- [FIX] On a fresh install the admin ACL rights were not properly set up which required a service restart to get fixed.
- [FIX] Force admin password change on upgrade if the old hash is SHA-1.
- [FIX] minor fixes for uWSGI support.
- [FIX] Fix temporary directories not removed after package import or edit.
- [FIX] Fix duplicated auth\_module\_ad.py module in bad waptwaptenterprise directory on windows waptserver.
- [IMP] Warning of wapt licence expiration message changed from 14 days to 60 days before expiration.
- [FIX] Fix broadcast for wakeonlan.
- [FIX] fix additional server password issues when non ascii character.

### Library changes

- [UPD] Update OpenSSL binary from 1.0.2r to 1.0.2u.
- [UPD] Update Python4Delphi lib to 20201020 release.
- [UPD] Build now with Lazarus 2.0.8 and FPC 3.0.4.

### 81.3.7 WAPT-1.8.2.7269 (2020-06-16)

hash : 757cdc76

- [FIX] Fix db schema upgrade script for upgrade from WAPT version 1.8.1-6742. Fresh 1.8.2 installation or upgrade from 1.7 or from 1.8.0 or 1.8.1-6758 shouldn't have the issue.
- [IMP] Add key for `install_msi_if_needed`.
- [FIX] Fix for `no_fallback` for `waptwua` (**Enterprise** only).

### 81.3.8 WAPT-1.8.2.7267 (2020-06-12)

hash : 46f40312

- [FIX] Fix db schema upgrade script for upgrade from WAPT version 1.8.1-6742. Fresh 1.8.2 installation or upgrade from 1.7 or from 1.8.0 or 1.8.1-6758 shouldn't have the issue.

### 81.3.9 WAPT-1.8.2.7265 (2020-06-11)

hash : 339f1996

This is mostly a bugfix release. Support for Linux and Mac clients has also been greatly improved.

### Notable enhancements

- [IMP] improve support for WaptAgent on Linux and Mac. Now the support is almost identical on Windows, Linux and MacOS (all versions) :
  - waptagent installation as a service with kerberos registration.
  - waptselfservice gui available on the 3 platforms (note : support for the latest version of MacOS, Catalina, is expected for 1.8.3).
  - waptexit (on Linux an Mac it is not yet started on system shutdown, it can be triggered by a scheduled task).
  - session-setup for configuring user sessions.
  - send messagebox to users and propose upgrades (**Enterprise** only).
  - OU handling (**Enterprise** only).
  - waptselfservice authentication can be delegated to the waptserver (**Enterprise** only).
  - better *setuptools* coverage.
- [IMP] new supported platforms. Now WAPT for linux (server and agent) and MacOS (agent only) supports :
  - Ubuntu 18.04 and 20.04;
  - Debian 8, 9 and 10;
  - Centos7 (CentOS 8 as a preview);
  - MacOS Sierra, HighSierra, Mojave (note : support for MacOS Catalina expected for WAPT 1.8.3).
- [IMP] streamlining of development environment for packaging on Linux using VSCode.
- [FIX] better handling of websocket cleanup when a host is not properly registered. Should improve stability on large WAPT installations.
- [IMP] selfservice can now be configured for external authentication for desktops that are not in a AD Domain.
- [IMP] selfservice users can now authenticate on waptserver even when out of the corporate network.

- [IMP] The session setup is run for all packages immediately after upgrade or install, so that new packages are already configured in the context of each logged in users (no need to logout / login) (**Enterprise** only).
- [IMP] If secondary repositories are defined in `waptconsole.ini`, additional packages can be selected when editing hosts, groups or self-service packages.
- [IMP] When editing group or self-service packages, one can define the Target OS of the package.
- [IMP] Remote message to logged in users is using the same custom dialog box for Windows, Linux and macOS.
- [IMP] Remote message to logged in users can display the same custom logo as self-service (**Enterprise** only).
- [IMP] The IP/Subnet match in repository access rules is based on the « main IP » of the host (source IP from which the host is reaching the server, if the server is public, this is usually the external IP of the router) (**Enterprise** only).
- [IMP] Added Remote host Shutdown and remote host Reboot from Waptconsole if enabled in `wapt-get.ini` (`allow_remote_shutdown` and `allow_remote_reboot`) (**Enterprise** only).
- [IMP] Add a *no fallback* checkbox in repositories access rule to prevent host using main repository in case secondary ones are not reachable (when main repository bandwidth is limited, having all hosts reaching the main repository can slow down access to the main site) (**Enterprise** only).
- [FIX] Make sure WUA install task are executed after packages install (**Enterprise** only).

## Other enhancements

- [IMP] Cmd Console is hidden when session-setup is running, to limit annoyance for users.
- [IMP] WUA direct download option in `waptconsole` (**Enterprise** only).
- [IMP] can now use microsoft url for WUA in rules (**Enterprise** only).
- [FIX] Improved background icons loading in self-service.
- [FIX] better inventory of `lastboottime` and `get_domain_info`.
- [FIX] better handling of other local install of Python on client computer (eg. conflict with local Anaconda Python installation).
- [IMP] allows to have multiple private repo content displayed in `waptconsole`.
- [IMP] remote repository : it is now possible to prevent a fallback.
- [FIX] better handling of icons in selfservice.
- [IMP] improved support for VSCode.
- [FIX] better handling of ipv6 in console and inventory.
- [IMP] `wapt_admin_filter` : local admin can be filtered out like normal user in selfservice.
- [IMP] add a larger support for setuphelpers on macOS.
- [FIX] `waptserver` logs are properly redirected to `/var/log/waptserver.log`.
- [FIX] package caching : packages are deleted after each successful installation (rather than at the end of the whole upgrade) to better keep local disk space.
- [IMP] allows usage of url for changelog in control file.
- [IMP] better support for Windows Update download directly from Microsoft if WAPTServer is not reachable.
- [FIX] better handling of upgrade from Community version to Enterprise version.
- [IMP] improved local store skin and translations.
- [FIX] bugfixes and minor gui improvements.

## Library changes in WAPT-1.8.2.7165

- [CHANGE] replaced `python-ldap` with `ldap3`.
- [FIX] upgraded `ujson` on waptagent and waptserver on Linux.

### Removed features with WAPT-1.8.2.7165

- [REMOVED] autoconfiguration of repositories based on SRV DNS fields (it was not working anymore anyway).

### Caveats when using WAPT-1.8.2.7165

- [CAV] WaptExit is not run automatically on shutdown on Linux or MacOS (current issue with **systemd** / launched integration).
- [CAV] WaptTray is not yet available on Linux and macOS.
- [CAV] MacOS Catalina is supported by the WaptAgent, however WAPTSelfService and WaptExit are not yet supported.

### 81.3.10 WAPT-1.8.2.7265 RC2 (2020-05-29)

hash git : 339f1996

**Avertissement :** This is a Release Candidate version for testing and evaluation only and should not be installed on production system.

This is mostly a bugfix release. Support for Linux and Mac clients has greatly improved.

### Notable enhancements over 1.8.2 RC1

- [IMP] the session setup in run for all packages immediately after upgrade or install, so that new packages are already configured in the context of each logged in users (no need to logout / login) (**Enterprise** only).
- [IMP] if secondary repositories are defined in waptconsole.ini, additional packages can be selected when editing hosts, groups or self-service packages.
- [IMP] when editing group or self-service packages, one can define the target OS of the package.
- [IMP] remote message to logged in users is using the same custom dialog box for windows, linux and macOS.
- [IMP] remote message to logged in users can display the same custom logo as self-service (**Enterprise** only).
- [IMP] the IP / Subnet match in repository access rules is based on the *main IP* of the host (source IP from which the host is reaching the server, if the server is public, this is usually the external IP of the router) (**Enterprise** only).
- [IMP] added remote host shutdown and remote host reboot from Waptconsole if enabled in wapt-get.ini (`allow_remote_shutdown` and `allow_remote_reboot`) (**Enterprise** only).
- [IMP] added a *no fallback* checkbox in repositories access rule to prevent hosts using main repository in case secondary repositories are not reachable (when main repository bandwidth is limited, having all hosts reaching the main repository can slow down access to the main site) (**Enterprise** only).
- [FIX] make sure WUA install task are executed after packages install (**Enterprise** only).

### Other enhancements over 1.8.2 RC1

- [IMP] cmd Console is hidden when session-setup is running, to limit annoyance for users.
- [IMP] WUA direct download option in waptconsole (**Enterprise** only).
- [IMP] can now use Microsoft url for WUA in rules (**Enterprise** only).
- [IMP] improved background icons loading in self-service.



## Removed features

None

## Caveats

Same as RC1

### 81.3.11 WAPT-1.8.2.7165 RC1 (2020-05-29)

hash git : 1387b38f

**Avertissement :** This is a Release Candidate version for testing and evaluation only and should not be installed on production system.

This is mostly a bugfix release. Support for Linux and macOS clients has greatly improved.

#### Notable enhancements in WAPT-1.8.2.7165 RC1

- [IMP] improve support for WaptAgent on Linux and Mac. Now the support is almost identical on Windows, Linux and MacOS (all versions) :
- waptagent installation as a service with kerberos registration.
- waptselfservice gui available on the 3 platforms (note : support for the latest version of MacOS, Catalina, is expected for 1.8.3).
- waptexit (on Linux and Mac it is not yet started on system shutdown, it can be triggered by a scheduled task).
- session-setup for configuring user sessions.
- send messagebox to users and propose upgrades (**Enterprise** only).
- OU handling (**Enterprise** only).
- waptselfservice authentication can be delegated to the waptserver (**Enterprise** only).
- better setphelpers coverage.
- [IMP] add new supported platform. Now WAPT for linux (server and agent) and MacOS (agent only) supports :
  - Ubuntu 18.04 and 20.04;
  - Debian 8, 9 and 10;
  - Centos7 (CentOS 8 as a preview);
  - MacOS Sierra, HighSierra, Mojave (note : support for MacOS Catalina expected for WAPT 1.8.3);
- [IMP] streamlining of development environment for packaging on Linux using VSCode.
- [FIX] better handling of websocket cleanup when a host is not properly registered. Should improve stability on large WAPT installation.
- [IMP] selfservice can now be configured for external authentication for desktops that are not in an Active Directory Domain.
- [IMP] selfservice users can now authenticate on selfserver even when out of the corporate network.

### Other enhancements in WAPT-1.8.2.7165 RC1

- [FIX] better inventory of `lastboottime` and `get_domain_info`.
- [FIX] better handling of other local install of Python on client computer (eg. conflict with local Anaconda Python installation).
- [IMP] allows to have multiple private repo content displayed in `waptconsole`.
- [IMP] remote repository : it is now possible to prevent a fallback.
- [FIX] better handling of icons in `selfservice`.
- [IMP] improved support for VSCode.
- [FIX] better handling of ipv6 in console and inventory.
- [IMP] `wapt_admin_filter` : local admin can be filtered out like normal user in `selfservice`.
- [IMP] add a larger support for `setuptools` on macOS.
- [FIX] `waptserver` logs are properly redirected to `/var/log/waptserver.log`.
- [FIX] package caching : packages are deleted after each successful installation (rather than at the end of the whole upgrade) to better keep local disk space.
- [IMP] allows usage of url for changelog in control file.
- [IMP] better support for Windows Update download directly from Microsoft if WAPTServer is not reachable.
- [FIX] better handling of upgrade from Community version to Enterprise version.
- [IMP] improved local store skin and translation.
- [FIX] bugfixes and minor gui improvements.

### Library changes in WAPT-1.8.2.7165 RC1

- [REF] replaced `python-ldap` with `ldap3`.
- [FIX] upgraded `ujson` on `waptagent` and `waptserver` on Linux.

### Removed featured with WAPT-1.8.2.7165 RC1

- autoconfiguration of repositories based on SRV DNS fields (it was not working anymore anyway).

### Caveats when using WAPT-1.8.2.7165 RC1

- [CAV] `WaptExit` is not run automatically on shutdown on Linux or MacOS (current issue with `systemd` / launched integration).
- [CAV] `WaptTray` is not yet available on Linux and MacOS.
- [CAV] MacOS Catalina is supported by the `WaptAgent`, however `WAPTSelfService` and `WaptExit` are not yet supported.

## 81.3.12 WAPT-1.8.1-6758 (2020-03-06)

(hash bb93ce41)

On server :

- [REF] refactoring for `postconf.py` / remove old migration from MongoDB ;
- [REF] refactoring for `winsetup.py` / create now a `dhparam` for `nginx` on Windows ;
- [REF] refactoring for repositories : change `repo_diff` by `remote_repo_diff` / add param `remote_repo_websockets` (by default to True) on server ;
- [IMP] disable cache on `nginx` for Windows and Linux on `wapt` packages / exe ;

On agents :

- [REF] change param `waptservice_admin_auth_allow` by `waptservice_admin_filter` ;
- [REF] delete `resync` functions for remote repo ;
- [IMP] param `local_repo_sync_task_period` by default to « 2h » ;

- [FIX] wapt-get / waptservice debug when download a package on linux when not sudo ;
  - [FIX] fix for **plist** in macOS ;
  - [IMP] can now have relative path for packages/directories in **wapt-get** ;
  - [IMP] templates have by default `setup_uninstall / update` etc. . .
  - [IMP] improvements with templates for `vscode` ;
- On waptconsole :
- [IMP] add possibility of template packages for `deb / rpm / pkg` ;
  - [FIX] Fix for `msi, exe, etc` in PackageWizard explorer ;
  - [IMP] Can now choose `editor_for_packages` directly in waptconsole config ;
  - [UPD] Some cosmetic / translations improvements for GUI to deploy waptagent ;

### 81.3.13 WAPT-1.8.1-6756 (2020-02-17)

(hash 43394f3b)

#### Bug fixes and small improvements

- [IMP] waptconsole : improve the refresh of hosts grid when a lot of hosts are selected (improved by a factor of around 5)
- [FIX] waptserver Database connections management : don't close DB on teardown as it should not occur, and seems to trigger some issue when triggering a lot of tasks on remote hosts (error db is closed)
- [FIX] waptconsole : Don't « force » install when triggering the upgrade on remote hosts, to avoid reinstalling softwares when already up to date.
- [IMP] use `ldap auth` only if session and admin fail (avoid waiting for timeout when ldap is not available but one wants to login with plain admin user) ;
- [FIX] wapt-get upload : encode user and password in `http_upload_package` to allow non ascii in admin password ;
- [IMP] waptconsole : Disable auto search on keywords ;
- [IMP] use `DMI System_Information.Serial_Number` information for `serialnr Host` field instead of `Chassis_Information.Serial_Number` because `System_Information` is more often properly defined ;
- [IMP] waptconsole : add `uuid` in the list of searched fields when only "host" is checked in filters ;
- [IMP] `nginx` config : disable caching ;
- [IMP] fixes for **vscode** project template ;

### 81.3.14 WAPT-1.8.1-6742 (2020-02-12)

(hash 80dbdbe7)

#### Major changes

- waptconsole : Added a page to show packages install status summary (merge) of all selected hosts, grouped by package, version, `install status`, with count of hosts ;

Context menu allow to apply selectively the pending actions. On enterprise, one can apply safely the updates (only packages for which there is no running process on client side) ;

- Prevent users from saving a host package if targeted host(s) do not accept their personal certificate. (Checked on waptconsole when editing / mass updating host packages, and on server when uploading packages) ;

The personal certificate file `.crt` must contain at first the personal certificate, followed by the issuer CA certificates, so that wapt can rebuild the certificate chain and check intersection with host's trusted certificates ;

### Important note about SSL client side authentication

In your nginx configuration, be sure to reset the headers `X-Ssl-Authenticated` and `X-Ssl-Client-DN` as waptserver *trusts* these headers if ssl client side auth is enabled in `waptserver.ini`;

If SSL client side auth is setup these headers can be populated by `proxy_set_header` with result of `ssl_verify_client` as explained in [./wapt-security/security-configuration-certificate-authentication.html#enabling-client-side-certificate-authentication](#);

### Fixes and detailed changelog

- Security fix : update waitress module to 1.4.3 (CVE-2020-5236);
- Security fix : blank `X-Ssl*` headers in default **nginx** templates;
- Fix : regression : **kerberos register\_host** did not work anymore;
- On server, `:file :"<repository root>/wapt/ssl"` dir is moved automatically on winsetup / postconf to (per default) `:file :"<repository root>/ssl"`, a `/ssl` location is added;

This `/ssl` should be accessible from clients at the location specified by the server parameter `clients_signing_crl_url` (in `waptserver.ini`);

- Improved logs readability. Log count of used DB connections from pool on waptserver to troubleshoot DB connection issues. Log level can be specified by subcomponent with `loglevel_waptcore`, `loglevel_waptserver`, `loglevel_waptserver.app`, `loglevel_waptws`, `loglevel_waptdb` defined in `waptserver.ini`;
- Reworked explicit DB Open/close on waptserver to not get a DB connection from pool if not useful. It prevents exhaustion of DB connections;
- `waptwinsetup` : don't create unused directories `wapt-group` and `waptserverlog`;
- Added `.msu` and `.msix` extensions for Package wizard setup file dialog;
- Fallback with `os._exit(10)` for waptservice restart. Added a handler in **nssm.exe** configuration to honor the restart;
- Increased waitress threads to 10 on waptservice;
- Lowered the default number of pooled DB connections (`db_max_connections`) to 90, to be lower than postgresql default of 100;
- `waptserver` : allow kerberos or ssl auth check in waptserver only if enabled in `waptserver.ini` config file;
- `waptconsole` : Allow update of host package only if user certificate is actually allowed on the host (based on last update of host status in database);
- `waptconsole / build waptagent` : checkbox to specify to include or not non certificate authority certificates in build. The normal setup would be to uncheck this, to not deploy non CA certificates, on wapt root CA;
- [IMP] Add and option to disable automatic hiding of panels...
- [IMP] Add explicit AllowUnauthenticatedRegistration task to waptserversetup windows
- `waptsetup` : Remove explicit VCRedistNeedsInstall task. Use `/VCRedistInstall=(0/1)` if you need to force install or force not install `vcredist VC_2008_SP1_MFC_SEC_UPD_REDIST_X86`;
- [FIX] **wapt-get.exe** : use `wapt-get.ini` for `:command : "scan-packages"` and `:command : "update-packages"` wapt-get actions;
- [FIX] **wapt-get** : auth asked when checking if server is available (ping) and client ssl auth is enabled;
- [IMP] WAPT client : if client ssl auth failed with http error 400, retry without ssl auth to be able to ask for new certificate signing;
- [FIX] waptserver register behavior : revert over rev 6641 : sign host certificate if an authenticated user is provided or data is signed with a key which can be verified by existing certificate in database for this host uuid;
- [IMP] waptserver register behavior : when receiving 401 from server when registering, retry registering without ssl auth;
- [IMP] wapt client : be sure to have proper host private key saved on disk when receiving signed certificate from server;
- [IMP] waptconsole : advanced filters for selected host packages status. Filter on *Install status* and *Section + keyword*. *Pending* button to show only pending installations / removes;
- [ADD] wapt-get make-template / edit package : Add `.vscode` directory. Add template project for vscode;
- [FIX] waptconsole : fix ssl auth for mass package dependencies / conflicts updates;
- [FIX] waptconsole : fix import packages from external repos with ssl auth;
- [IMP] backports from master :

- target OS in import packages;
  - choose editor for packages in linux in cmdline;
  - [IMP] backports from master :
  - refactoring for `HostCapabilities.waptos`;
  - add new `target_os` unix for mac and linux;
  - so `target_os` : windows, darwin (for mac), linux or unix;
  - [FIX] `WAPT.wapt_base_dir`;
  - [FIX] `makepath` in linux/macOS;
  - [IMP] refactoring / fixes for `setuphelpers`;
  - [FIX] for `rights_to_check` in repo-sync client;
  - [FIX] for `repo-sync`;
  - [ADD] two `setuphelpers` for linux : `type_debian` and `type_redhat`
- indent the local `sync.json`;
- [IMP] use `get_os_version` and `windows_version_from_registry` instead of `windows_version`;
  - [IMP] use `windows_version_registry` for `get_os_version` on windows;
  - [IMP] backport `host_capabilities.os` from master
  - [FIX] for **make-template** for malformed `.exe` installer;
  - [ADD] automatic maintenance of a CSR for client auth certificates signed by server :
  - default CSR lifetime to 30 days;
  - check renewal of client cert CSR every hour;
  - added a parameter for the next update time of `crl`;
  - added `clients_signing_crl_url`, `clients_signing_crl_days`, `known_certificates_folder` `waptserver` parameters;
  - added a `/ssl` location in `nginx` templates;
  - added `crl_urls` in client auth signed certificates;
  - added a scheduled task to renew server side `crl`;
  - added `clients_signing_crl` `waptserver` parameter to add client cert to server `crl` when host is unregistered;
  - added **revoke\_cert** method to `SSLCRL` class;
  - added a `authorityKeyIdentifier` to the client auth CSR;
  - force restart if windows task is broken;
  - `waptservice` : use `sys._exit(10)` to ask **nssm** to restart service in case of unhandled exception in `waptservice` (loops, etc.);
  - `wapt` client : don't log / store into db `Wapt.runstatus` if not changed;
  - `waptserver` `postconf` : fix for rights on some `wapt` directories;
  - Add mutual conflicts to `deb/rpm` packages for `waptagent/waptserver` to avoid simultaneous install;

### 81.3.15 WAPT-1.8.0-6641 (2020-01-24)

(hash 3dbb3de8)

#### Major changes

- [ADD] client Agent for Linux Debian 8, 9, 10, Linux Centos 7, Ubuntu 18, 19 and MacOS. The packages are named wapt-agent and available in <https://wapt.tranquil.it/wapt/releases/latest/>;
- [IMP] repository access rules defined in waptconsole. Depending of client IP, site, computername, one can define which secondary repository URL to use (**Enterprise** only);

#### As a consequence, the DNS query method (with SRV records) is no more supported for repositories

- [IMP] the package and signature process has been changed to be compatible with **python3**. Serialization of dict is now sorted by key alphabetically to be deterministic across python versions. WAPT agents prior to version 1.7.1 will not be able to use new packages. (see git hash SHA-1 : f571e55594617b43ed83003faeef4911474a84db);
- [NEW] a WAPT agent can now be declared as a secondary remote repository. Integrated syncing with main server repository is handled automatically. (**Enterprise** only);
- [NEW] waptconsole can now run without elevated privileges. The build of waptagent / waptupgrade package are done in a temporary directory. **When editing a package from waptconsole, program `PyScripter` should be launched with elevated privileges;**

---

**Note :** One could deploy the agent with GPO without actually rebuilding a waptagent. Command line options are available on stock waptsetup-tis.exe to configure repo url (/repo\_url=), server url (/wapt\_server=), server certificate bundle location (/CopyServersTrustedCA=), packages certificates checking (/CopyPackagesTrustedCA=), /use\_random\_uuid, /StartPackages, /append\_host\_profiles, /DisableHiberBoot, /waptaudit\_task\_period;

---

Some options are still missing and may be added in a future release;

- [IMP] package filename now includes a hash of package content to make it easier to check if download is complete and if package has been scanned (improved speed for large number of packages);
- [SEC] the WAPT admin password must be regenerated (with postconf); if it is not *pbkdf2* based. See in your waptserver.ini file, wapt\_password must start with **\$pbkdf2-**;

#### Fixes and detailed changelog

- [SEC] waptagent can optionally be digitally signed, if (1) Microsoft **signtool.exe** is present in <wapt>utils` and (2) if there is a pkcs#12 :mimetype:.p12` file with the same name as the personal certificate .crt file, and (3) the certificate is encrypted with the same password;
- [IMP] wapt-get.py can be run on linux and macos in addition to windows;
- [IMP] waptconsole host's packages status reporting : now displays current version with *NEED-UPGRADE*, *NEED-REMOVE*, *ERROR* status and future version with *NEED-INSTALL* status;

The status is stored in server's DB HostPackagesStatus so it can be queried for reporting;

- [IMP] setuphelpers : there now different setuphelpers for each operating system family;
- [ADD] waptconsole : added an action to safely trigger upgrades on remote hosts only if associated processes (impacted\_process control attribute) are not running, to avoid disturbing users (**Enterprise** only);
- [ADD] **wapt-get --service upgrade** : added handling of **--force**, **--notify\_server\_on\_start=0/1**, **notify\_server\_on\_finish=0/1** switches;
- [IMP] package signature's date is now taken in account when comparing packages;
- [ADD] **host\_ad\_site** key in [global] in wapt-get.ini to define a *fake* Active Directory site for the host;
- [ADD] waptconsole / packages grid : if multiple packages are selected, the associated *show clients* grid shows the status of packages for all selected clients (**Enterprise** only);

- [ADD] waptagent build : added checkbox to enable repository rules lookup when installing agent (**Enterprise** only);
- [ADD] waptconsole / import packages : don't reimport existing dependencies. Checkbox to disable import of dependencies;
- [IMP] wapt-scanpackages speed optimizations : don't re-extract certificates and icon for skipped package entries. use md5 from filename if supplied when scanning.
- [FIX] waptexit : fix arguments to waptexit for `only_if_not_process_running` and `install_wua_updates` (bool);
- [FIX] waptagent / waptwua fix wapt wua enabled setting reset to *False* when upgrading with waptagent and `enabled=don't touch`;
- [FIX] waptserver / waptwua repository : all cabs files are now in root directory instead of microsoft original file tree. The files are moved when upgrading to 1.8;
- [IMP] waptupgrade package : increment build number if building a new waptagent of the same main wapt version;
- [NEW] waptserver parameter `trusted_signers_certificates_folder` :  
Path to trusted signers certificate directory. If defined, only packages signed by this trusted CA are accepted on the server when uploading through server;
- [NEW] waptserver parameter `remote_repo_support` : if true, a task is scheduled to scan repositories (`wapt`, `waptwua`, `wapt-hosts`) that creates a `sync.json` file for remote secondary repositories;
- [IMP] when building waptagent, don't include non CA packages certificates by default in waptagent. A checkbox is available to still enable non CA certificates to be scanned and added;
- [IMP] when building waptagent, one can add or remove certificates in the grid with `Ctrl+Del` or drag and drop;
- [FIX] waptconsole / host packages status grid : fixed F5 refresh;
- [IMP] waptconsole / build agent : build an enterprise agent even if no valid licence (**Enterprise** only);
- [FIX] `forced_update_on` control attribute : don't take into account for `next_update_on` if in the past;
- [IMP] waptconsole : try to accept waptserver password with non ASCII characters;
- [REMOVED] waptstarter : remove *sofle* from default host profile;
- [IMP] waptagent build : rework of server certificate path relocation when building / installing;
- [SEC] don't sign agent certificate if no valid human authentication (admin, passwd or ldap) or kerberos authentication has been provided :
  - be explicit on authentication methods;
  - store registration authentication method in db only if valid human authentication or kerberos authentication has been provided;
  - when registering, be sure we trust an already signed certificate with CN matching the host;
  - store the signed host certificate in server DB on proper registration;
- [IMP] some syntax preparation work for future python3;
- [IMP] some preparation work for detailed ACL handling (**Enterprise** only);
- [FIX] don't enable client ssl auth by default in waptserver as nginx reverse proxy server is perhaps misconfigured;

### Python libraries / modules updates

- use **waitress** for waptservice wsgi server instead of unmaintained **Rocket`**;
- **Flask-SocketIO 3.0.1** -> **Flask-SocketIO 4.2.1**;
- **MarkupSafe 1.0** -> **MarkupSafe 1.1.1**;
- **python\_ldap-2.4.44** -> **python\_ldap-3.2.0**;

## 81.4 WAPT 1.7 and older

### 81.4.1 WAPT-1.7.4-6237 (2019-11-18)

(hash 1c00cefd)

- [FIX] waptserver : add fix to workaround `flask-socketio` bug (AttributeError : “Request” object has no attribute “sid”);
- [IMP] waptserver : be sure db is closed before trying to open it (for dev mode);
- [IMP] waptserver : add logs messages when an exception message is sent back to the user;

### 81.4.2 WAPT-1.7.4-6234 (2019-11-14)

(hash ad237eee)

- [IMP] waptserver : upgrade **peewee** DB python module to 3.11.2. Explicit connection handling to DB to track potential limbo connections (which could lead to db pool exhaustion);
- [FIX] waptwua : trap exception when pushing WU to Windows cache to allow valid updates to be installed even if some could not be verified properly;

### 81.4.3 WAPT-1.7.4-6232 (2019-10-31)

(hash2090b0e6d52cecfb04f8fa4c279e7c0a0252d6e2)

- [FIX] **wapt-get session-setup** : fix bad print in **session\_setup**. Regression introduced in b30b1b1a550a4 (1.7.4.6229);

### 81.4.4 WAPT-1.7.4-6230 (2019-10-23) (not released)

(hash 391d382f)

- [IMP] return server git hash version and edition in `ping` and `usage_statistics`;
- [IMP] be sure to have `server_uuid` on windows when during setup;
- [FIX] `.git` partially included in built package `manifest`;

### 81.4.5 WAPT-1.7.4-6229 (2019-10-23)

(hash b30b1b1a)

- [FIX] 100% cpu load on one core on waptserver even when Idle;
- **python-engineio** upgrade to 3.10.0;
- **python-socketio** upgraded to 4.3.1;
- [IMP] don't try run **session\_setup** on packages which don't have one defined;
- [IMP] limit text output on console (for faster output);



### 81.4.6 WAPT-1.7.4-6223 (2019-10-15)

(hash 86ddea2d)

- [FIX] Newlines in packages installs logged output;
  - [FIX] Allow nonascii utf8 encoded user and password for server basic auth;
  - [UPD] waptconsole : Default package filtering to x64 and console locale to avoid mistakes when importing;
  - [IMP] waptconsole : increase default Port Socket listening test timeout (for rdp, remote service access etc..) to 3s instead of 200ms;
  - [IMP] waptconsole : sort OU by description in treeview :
- Right click changes current row selection in OU treeview ;
- [NEW] option to set waptservice\_password = **NOPASSWORD** in waptstarter installer;
  - [FIX] grid sorting for package / version / size of packages;
  - [FIX] don't create waptconsole link for starter;
  - [NEW] **wapt-scanpackages** : add an option to update the local packages DB table from Packages file index ;
  - [FIX] regression introduced in previous build : maturities = **PROD** and maturities = **“”** are equivalent when filtering allowed packages ;
  - [FIX] waptconsole : grid headers too small for highdpi ;
  - [UPD] waptupgrade package filename : keep old naming without *all arch* (for backward compatibility) ;
  - [IMP] waptservice\_timeout = **20** seconds now ;
  - [FIX] AD auth for waptconsole with non ASCII chars ;
  - [IMP] missing french translations for columns in *Import packages* grid ;
  - [FIX] be sure to terminate output threads in waptwinutils.run ;
  - [IMP] avoid showOnTop flickering for VisLoading ;
  - [IMP] setuphelpers.run\_powershell ! add \$ProgressPreference = **SilentlyContinue** prefix command ;
  - [SEC] waptservice : protect test of host\_cert date if file is deleted outside of service scope ;
  - [IMP] WaptBaseRepo class :
  - packages cache handling when repo parameters (filters...) are changed ;
  - allow direct setting of cabundle for WaptBaseRepo ;
  - keep a fingerprint of input config parameters ;
  - [UPD] set a fallback calculated package\_uuid value in database for compatibility with old package status reports ;

### 81.4.7 WAPT-1.7.4-6196 (2019-09-27)

(hash f9cb3ebd)

- [IMP] revert package naming of waptupgrade to previous one to ease upgrade from previous wapt ;
- [IMP] increase waptservice\_timeout to 20 seconds per default ;
- [FIX] AD auth when there are non ascii chars (encoding) ;
- [FIX] missing french translations for columns in Import packages grid ;
- [IMP] set a fallback calculated package\_uuid in database for old package without package\_uuid attribute in db status report ;
- [NEW] **wapt-scanpackages** : add an option to update the local Packages DB table from Packages file index ;
- [NEW] option to filters maturities ;

## 81.4.8 WAPT-1.7.4-6192 (2019-09-17)

(hash 3e00ac6688)

- [SEC] update python modules **python-engineio** and **werkzeug** to fix vulnerability [CVE-2019-14806](#)
- GHSA-j3jp-gvr5-7hwq
- [UPD] Python modules :
    - **eventlet 0.24.1** -> **eventlet 0.25.1**;
    - **flask 1.0.2** -> **flask 1.1.1**;
    - **greenlet 0.4.13** -> **greenlet 0.4.15**;
    - **itsdangerous 0.24** -> **itsdangerous 1.1.0**;
    - **peewee 3.6.4** -> **peewee 3.10**;
    - **python-socketio 1.9.0** -> **python-socketio 4.3.1**;
    - **python-engineio 3.8.1** -> **python-engineio 3.9.3**;
    - **websocket-client 0.50** -> **websocket-client 0.56**;
  - [UPD] default `request_timeout = 15s` for client websockets;
  - [FIX] when building packages, excluded directories (for example `.git` or `.svn`) were still included in `manifest` file;
  - [UPD] don't canonicalize package filenames by default when scanning server repository to ease migration from previous buggy wapt;
  - [FIX] package filename not rewritten in `Packages` when renaming package;
  - [NEW] **wapt-scanpackages** : added explicit option to trigger rename of packages filenames which do not comply with canonic form;
  - [NEW] **wapt-scanpackages** : added option to provide proxy;
  - [UPD] return **OK** by default in package's audit skeleton;
  - [IMP] waptconsole cosmetic : minheight 18 pixels for grid headers
  - [FIX] waptserver database model : bad default datatype in `model.py` for `created_by` and `updated_by` (were not used until now);
  - [FIX] `ensure_unicode` for `.msi` output : try `cp850` before `utf16` to avoid Chinese garbage in run output;
  - [NEW] added `connected_users` to `hosts_for_package` provider;
  - [FIX] use **win32api** to get local connected IPV4 IP address instead of socket module. In some cases, socket can't retrieve the IP;
  - [FIX] **wapt-get unregister** command not working properly;
  - [NEW] Waptselfservice : added option in `wapt-get.ini` to disable unfiltered packages view of local admin;
  - [IMP] Waptselfservice : 4K improvements;
  - [FIX] Waptselfservice :
    - packages *restricted* were shown in selfservice / now corrected;
    - if the repo have no packages segmentation error / now corrected;
    - if the repo have changed segmentation error / now corrected;

### 81.4.9 WAPT-1.7.4.6165 (2019-08-02)

(hash f153fab4)

#### Improvements

- [NEW] added **unregister** action to wapt-get;
- [UPD] improvements with the alt logo in the self-service;

#### Changes

- [UPD] use version to build the package name of unit, groups and profile type package, like for base packages;
- [UPD] added logs to **uwsgi**;

#### Fixes

- [FIX] bugfixes with the icons of the app self-service;
- [FIX] bugfixes with the logos in the self-service;
- [UPD] waptexit : don't cancel tasks on CloseQuery;
- [UPD] patch `server.py` earlier to avoid *execute cannot be used while an asynchronous query is underway*;
- [FIX] fix waptexit doing nothing if `allow_cancel_upgrade = 0` and `waptexit_disable_upgrade = 0`;
- [FIX] fix issue with merge of wsus rules (can cause memory errors if more than one wsus package is applied on a host) (**Enterprise** only);
- [FIX] fix wua auto `install_scheduling` issue;
- [FIX] waptexit : add a watchdog to workaround some cases where it hangs (threading issue?);

### 81.4.10 WAPT-1.7.4.6143 (2019-06-25)

(hash da870a2c)

#### Improvements

- [IMP] wapt self service application is now fully usable. It is available in `<wapt>waptself.exe`;
- [ADD] option to set a random UUID instead of BIOS UUID at setup. This is to workaround for bugged BIOS with duplicated ids;
- [IMP] better Sphinxdocs for WAPT Libraries;

#### Changes

- [UPD] behavior change : Use computer FQDN from tcpip registry entry (first NV Hostname key) then fixed domain then DHCP;
- [FIX] inverted Zip and signature steps in package build operations to workaround issue with Bad Magic Number when signing already zipped big packages;
- [NEW] Add `use_ad_groups` wapt-get `[global]` parameter to activate groups from AD (this is a time consuming task, so better not activate it...);

### Fixes

- [FIX] appendprofile infinite loop during setup;
  - [FIX] read forced uuid from `wapt-get.ini` earlier to avoid loading a bad host certificate in memory if changing from bios uuid to forced uuid;
  - [FIX] setting `use_random_uuid` in `waptagent.iss`;
  - [FIX] waptstarter setup : force deactivate server, hostpackages;
  - [FIX] include waptself in waptstarter, don't include innosetup in waptstarter;
  - [FIX] `ensure_unicode` : add `utf16` decoding test before `cp850`;
  - [FIX] add `ensure_unicode` for tasks logs to avoid unicode decode errors in `get_tasks_status` callback;
  - [NEW] host status : add `boot_count` attribute;
  - [FIX] fix potential float / unicode error when scanning windows updates (**Enterprise** only);
  - [FIX] handles properly excluded files in package signatures;
  - [FIX] waptexit : avoid some work after checking if waptservice is running if it is not running;
  - [FIX] a case where WAPTLocalJsonGet could loop forever if auth fails;
  - [FIX] `setup.pyc` in `manifest` but not in zipped package :
  - exclude exactly [`".svn"`,"`.git`", "`.gitignore`", "`setup.pyc`"] when signing and zipping;
  - **inc\_build** before signing;
  - [UPD] add `use_ad_groups` setting in waptagent build. Default to `False` (**Enterprise** only);
  - [FIX] better detection of `waptbasedir` for `python27.dll` loading;
  - [FIX] allow to sign source package directory to workaround a bug in python zipfile (bad magic number);
  - [NEW] added a `htpasswd` password file method for restricted access to only **add\_host** method :
- allows **add\_host** if provided host certificate is already signed by server and content can be verified;
- [FIX] **wapt-get.exe** crash with « can not load... » when python 3.7 is installed from MS store;
  - [FIX] load `private_dir` conf parameter earlier;
  - [UPD] put a `rnd-` in front of randomly generated uuid;
- added a checkbox to use random uuid (if not already defined in `wapt-get.ini`);
- [UPD] SSL CA certifi library;
  - [IMP] utf8 decode user /password in localservice authentication;
  - [UPD] allow authentication on local waptservice with token;
  - [NEW] filter packages on hosts based on the `valid_from` and `valid_until` control attributes;
- force update sooner if `valid_from` or `valid_until` or `forced_install_on` is sooner than regular planned `update_period`;
- [FIX] events reporting from service tasks;
  - [FIX] **waptexit** not closing of writing for running tasks but auto upgrade has been disabled;
  - [ADD] added `waptexit_disable_upgrade` option to **waptexit** to remove the triggering of upgrade from waptexit, but keep the waiting for pending and running tasks :
- “running\_tasks” key in waptservice checkupdates.json. Was not reflecting an up to date state;
- [NEW] add new packages attributes : `name`, `valid_from`, `valid_until`, `forced_install_on`;
  - [FIX] regression on *profile* packages not taken in account;

### 81.4.11 WAPT-1.7.4.6082 (2019-05-20)

(hash 38e08433)

## Fixes

- [FIX] **waptexit** not closing if waiting for running tasks but auto upgrade has been disabled;
- [FIX] events reporting from service's tasks;

## Updated

- [ADD]] new packages attributes : `name`, `valid_from`, `valid_until`, `forced_install_on`;
- [ADD] `waptexit_disable_upgrade` option to **waptexit** to remove the triggering of upgrade from waptexit, but keep the waiting for pending and running tasks;
- [IMP] added `running_tasks` key in waptservice heckupgrades.json. Was not reflecting an up to date state.
- [IMP] waptself :
- early support of high DPI;
- loading of icons in the background;

### 81.4.12 WAPT-1.7.4.6078 (2019-05-17)

(hash 5b6851ae)

## Fixes

- [FIX] takes *profile* packages (AD based groups) into account (**Enterprise** only)

### 81.4.13 WAPT-1.7.4.6077 (2019-05-15)

(hash 4be40c534c4627)

## Fixes

- [FIX]] regression on waptdeploy unable to read current `waptversion` from registry;
- [FIX] be more tolerant to broken or inexistent *wmi* layer (for waptconsole on **wine** for example);

### 81.4.14 WAPT-1.7.4.6074 (2019-05-09)

(hash 95a146c002)

## Fixes and improvements over RC2

- [IMP] **waptself.exe** preview application updated. Loads icons in the background.

## Known issues :

- does not work with repositories behind proxies and client side auth;
- https server certificate is not checked when downloading icons);
- High DPI not handled properly;
- Cosmetic and ergonomic improvements still to come;
- [IMP] waptserver setup on windows : open port 80 on firewall in addition to 443;

- [IMP] waptserver on Debian. add *www-data* group to wapt user even if user wapt already exists;
- [IMP] waptserver on CentOS. add waptwua directory to SELinux `httpd_sys_content_t` context;
- [FIX] waptserver client auth : comment out `ssl_client_certificate` and `ssl_verify_client`;

By default because old client's certificate does not have proper `clientAuth` attribute (error http 400);

- [FIX] problem accessing to 32bit uninstall registry view from 32bit wapt on Windows server 2003 x64 and Windows server 2008 x64 :

it looks like it is not advisable to try to access the virtual `Wow6432Node` virtual node with disabled redirection ;

- [FIX] `setuptools_helpers_installed_softwares` regular expression search on name;

<https://github.com/tranquilit/WAPT/issues/7>

- [IMP] waptservice : for planned periodic upgrade, use single `WaptUpgrade` task like the one used in `websocket` ;
- [IMP] waptexit : cancel all tasks if closing waptexit form ;
- [FIX] wapt-get : wapt-get service mode with events : refactor using `uWAPTPollThreads` ;
- [FIX] **veyon** cli executable name updated ;
- [IMP] wapt-get : check `CN` and `subjectAltNames` in lowercase for **enable-check-certificate** action ;

(todo : doesn't take wildcard in account)

### 81.4.15 WAPT-1.7.4 RC2 (2019-04-30)

(hash 5ef3487)

#### Security

- upgrade **urllib3** to 1.24.2 for [CVE-2019-11324](#) (high severity);
- upgrade **jinjja2** to 2.10.1 for [CVE-2019-10906](#);

#### New

- [NEW] Wapt self service application preview ;

#### Improvements

- [IMP] propose to copy the newly created CA certificate to `ssl` local service dir, and restart waptservice. Useful for first time use ;

#### Fixes

- [FIX] `sign_needed` for `wapt-signpackages.py` ;
- [FIX] missing `StoreDownload` table create ;
- [FIX] bug in fallback `package_uuid` calculation. It didn't include the version ;

## 81.4.16 WAPT-1.7.4 RC1 (2019-04-16)

(hash 4cdcaa06c83b)

### Changes

- [UPD] handling of *subjectAltName* attribute for https server certificates checks in waptconsole (useful when certificate is a multi hostname commercial certificate). Before, only CN was checked against host's name;
- [UPD] client certificate auth for waptconsole;
- [UPD] versioning of wapt includes now the Git revision count;

### Details

- [FIX] replace openssl command line call with waptcrypto call to create tls certificate on linux server wapt install;
- [FIX] add dnsname *subjectAltName* extension to self signed waptserver certificate on linux wapt nginx server configuration;
- [FIX] pkcs12 export;
- [NEW] handling of *SubjectAlternativeName* in certificates for server X509 certificate check in addition to CN :

Added a *SubjectAltName* when creating self signed certificate on linux wapt nginx server in postconf;

For old installation, the certificate is not updated. It should be done manually;

- [FIX] fix **check\_install** returning additional packages to install which are already installed (when private repository is using `locale` or `maturities`):

Added missing attributes in waptdb.installed\_matching;

- [NEW] added client certificate path and client private key path for waptconsole access to client side ssl auth protected servers;
- [FIX] fix regression with **wapt-get edit <package>** :

made `filter_on_host_cap` a global property of Wapt class instead of a function parameter;

- [FIX] regression if there are spaces in OU name. Console was stripping space for <https://roundup.tranquil.it/wapt/issue911> and <https://roundup.tranquil.it/wapt/issue908>;
- [IMP] allow "0".."9", "A".."Z", "a".."z", "-", "\_", "=", "~", "," in package names for OU packages. Replaces space with ~ in package names and "," with "\_";
- [IMP] make sure we have a proper package name in packages edit dialogs;
- [IMP] waptservice config : allow `waptupdate_task_period` to be empty in `wapt-get.ini` to disable it in waptservice;
- [FIX] waptutils : fix regression on `wget()` if user-agent is overridden;
- [FIX] waptwua : fix an error in install progress % reporting for wua updates;
- [IMP] wapttray : refactor tray for consistency. Makes use of *uwaptollthreads* classes;
- [IMP] waptexit : some changes to try to fix cases when it does not close automatically;
- [IMP] build : add git Revcount (commit count) to exe metadata;
- [FIX] waptconsole : fix hosts for package grid not refreshed if not focused;
- [FIX] internal : use synapse `httpsend` for waptexit / wapt-get / wapttray local service http queries to workaround auth retry problems with **indy**;
- [ADD] **wapt-get.exe** : added `--locales` to override temporarily locales form `wapt-get.ini`;
- [ADD] **wapt-get.exe** : added *WaptServiceUser* and *WaptServicePassword* / *WaptServicePassword64* command line params :

fix timeout checking in checkopenport;

- [ADD] core : added logs for self-service auth;
- [ADD] waptservice : added /keywords.json service action;
- [ADD] waptservice : added filter keywords (csv) on packages.json provider;
- [IMP] waptconsole : replace tri-state checkbox by a radio group for wua enabled setting in *create waptagent* dialog;
- [IMP] waptservice local webservice : temporary workaround to avoid costly icons retrieval in local service;
- [FIX] simplify `installed_wapt_version` in waptupgrade package to avoid potential install issues;
- [IMP] waptconsole layout : anchors for running task memo;

- [FIX] Makefullyvisible for main form :  
avoid forms outside the visible area when disconnecting a second display ;
- [FIX] layout of tasks panel for Windows 10;
- [FIX] add `token_lifetime` server side (instead of using `clockskew` for token duration);
- [UPD] default unit **days** instead of **minutes** for wua scan download install and `install_delay` ;
- [ADD] optional export of key and certificate as PKCS12 file in `create key` dialog. (to check SSL client auth in browsers...);
- [FIX] `winsetup.py` fix for backslashes in **nginx** ;
- [FIX] `wapt-get` json output / flush error ;
- [IMP] cache `host_certificate_fingerprint` and issuer id in local db so that we don't need to read private directory to get `host_capabilities`. It allows to use **wapt-get list-upgrade** as normal user ;
- [UPD] don't make DNS query in `waptconsole` Login / `waptconfig` to avoid DNS timeout if domain dns server is not reachable ;
- [FIX] warning message introduced in previous revision when adding a new ini config on login (**Enterprise** only) ;
- [FIX] `waptwua` : handles redirect for `wsusscn2` head request (**Enterprise** only) ;
- [UPD] Report only 3 members on the `wapt_version` capability attribute ;
- [IMP] core : refactor `WaptUpgrade` task : check task to append and then append them to tasks queue in `WaptUpgrade.run` instead of doing it in caller code. Avoid timeout when upgrading ;
- [IMP] core : self service rules refactoring ;
- [IMP] core : notify server when audit on `waptupgrade` ;
- [IMP] core : fix `update_status` not working when old packages have no `persistent_dir` in db ;
- [IMP] core : tasks, events `waptservice` action : timeout in milliseconds instead of seconds for consistency ;

### 81.4.17 WAPT-1.7.3.11 (2019-03-25)

(hash 92ccb177d5c)

- [FIX] `waptconsole` : use repo specific ca bundle to check remote repo server certificate (different from main wapt repo) ;
- [FIX] `waptconsole` / hosts for packages : fixed F5 to do a local refresh ;
- [FIX] improved update performance with repositories with a lot of packages ;
- [FIX] improved `waptray` reporting :  
fix faulty inverted logic for `notify_user` parameter ;
- [FIX] `waptconsole` : fixed bad filtering of hosts for package (**Enterprise** only) ;
- [FIX] `waptexit` : fixed `waptexit` closes even if Running task if no pending task / no pending updates ;
- [FIX] `waptexit` : fixed potential case where `waptexit` remains running with high cpu load ;
- [FIX] `waptconsole` : fixed `HostsForPackage` grid not filtered properly (was improperly using `Search` expr from first page) ;
- [FIX] `waptservice` : None has no `check_install_is_running` error at `waptservice` startup ;
- [FIX] core : set `persistent_dir` and `persistent_source_dir` attributes on setup module for `install_wapt` ;
- [FIX] core : fixed bug in guessed `persistent_dir` for dev mode ;
- [FIX] core : fixed error resetting status of stuck processes in local db (`check_install_running`) ;
- [FIX] `waptservice` : trap error setting `runstatus` in db in tasks manager loop :  
Don't send `runstatus` to server each time it is set ;
- [UPD] core : define explicitly the `private_dir` of `Wapt` object ;
- [UPD] server : don't refuse to provide `authtoken` if FQDN has changed (this does not introduce specific risk as request is signed against UUID) ;
- [UPD] core : if `package_uuid` attribute is not set in package's `control` (old wapt), it is set to a reproducible hash when package is appended to local `waptdb` so we can use it to lookup packages faster (dict) ;
- [NEW] `waptconsole` : added audit scheduling setup in `waptagent` dialog (**Enterprise** only) :  
added `set_waptaudit_task_period` in `innosetup` installers ;
- [IMP] `setuphelpers` : add `win32_displays` to default wmi keys for report ;
- [IMP] server setup : create X509 certificate / RSA key for hosts ssl certificate signing and authentication during setup of server ;
- [IMP] `waptexit` : add sizeable border and icons ;
- [IMP] show progress of long tasks ;



- [IMP] waptservice : process update of packages as a task instead of waiting for its completion when upgrading (to avoid timeout when running upgrade waptservice task) :  
added `update_packages` optional (default True) parameter for upgrade waptservice action ;
- [NEW] added audit scheduling setup in waptagent compilation dialog (**Enterprise** only) ;
- [NEW] setuphelpers : added `get_local_profiles` setuphelpers ;
- [IMP] waptserver : don't refuse to provide authtoken for websockets auth if FQDN has changed ;
- [IMP] flush stdout before sending status to waptserver ;
- [IMP] waptcrypto handle alternative object names in CSR build ;
- [IMP] wapt-get : `--force` option on **wapt-get.exe** service mode ;
- [NEW] use client side authentication for waptwua too ;
- [CHANGE] server setup : nginx windows config : relocate logs and pid ;
- [ADD] added conditional client side ssl auth in nginx config ;
- [CHANGE] waptconsole : refactor `wget`, `wgets` `WaptRemoteRepo` `WaptServer` to use `requests.Session` object to handle specific ssl client auth and proxies :

**Be sure to set `privateKey` password dialog callback to decrypt client side ssl auth key ;**

- [IMP] waptcrypto : added `waptcrypto.is_pem_key_encrypted` ;
- [IMP] waptconsole : make sure waptagent window is fully visible ;
- [IMP] waptconsole : make sure Right click select row on all grids ;
- [ADD] waptconsole : import from remote repo : add certificate and key for client side authentication ;

## 81.4.18 WAPT-1.7.3.10 (2019-03-06)

(hash ec8aa25ef)

### Security

- [UPD] upgraded **OpenSSL** dlls to 1.0.2r for <https://www.cert.ssi.gouv.fr/avis/CERTFR-2019-AVI-080/> (moderate risk) ;

### New

- [IMP] much reworked wizard pages embedded in **waptserversetup.exe** windows server installer. Install of waptserver on Windows is easy again :
  - register server as a client of waptserver ;
  - create new key / certificate pair ;
  - build waptagent.exe and waptupgrade package ;
  - configure package prefix ;
- [NEW] if client certificate signing is enabled on waptserver (`waptserver.ini` config), the server will sign a CSR for the client when the client is first registered. See *Configuration de l'authentification par certificat côté client* .
- [NEW] wapt-get : added new command `create-keycert` to create a pair of RSA key / x509 certificate in batch mode. Self signed or signed with a CA key/cert :

**(options are case sensitive...)**

- option `/CommonName` : CN to embed in certificate ;
- options `/Email`, `/Country`, `/Locality`, `/Organization`, `/OrgUnit` : additional attributes to embed in certificate ;
- option `/PrivateKeyPassword` : specify the password for private key in clear text form ;
- option `/PrivateKeyPassword64` : specify the password for private key in base64 encoding form ;
- option `/NoPrivateKeyPassword` : ask to create or use an unencrypted RSA private key ;
- option `/CA = 1` (or 0) : create a certification authority certificate if 1 (default to 1) ;
- option `/CodeSigning = 1` (or 0) : create a code signing certificate if 1 (default to 1) ;

- option `/ClientAuth = 1` (or 0) : create a certificate for authenticating a client on a https server with ssl auth. (default to 1);
- option `/CAKeyFilename` : path to CA private key to use for signing the new certificate (defaults to `%LOCALAPPDATA%\waptconsole\waptconsole.ini [global] default_ca_key_path` setting);
- option `/CACertFilename` : path to CA certificate to use for signing the new certificate (defaults to `%LOCALAPPDATA%\waptconsole\waptconsole.ini [global] default_ca_cert_path` setting);
- option `/CAKeyPassword` : specify the password for CA private key in clear text form to use for signing the new certificate (no default);
- option `/CAKeyPassword64` : specify the password for CA private key in base64 encoding form to use for signing the new certificate (no default);
- option `/NoCAKeyPassword` : specify that the CA private to use for signing the new certificate is unencrypted;
- option `/EnrollNewCert` : copy the newly created certificate in `<wapt>ssl` to be taken in account as an authorized packages signer certificate;
- option `/SetAsDefaultPersonalCert` : set `personal_certificate_path` in configuration inifile [global] section (default `%LOCALAPPDATA%\waptconsole\waptconsole.ini`);
- [NEW] `wapt-get` : added new commands `build-waptagent` to compile a customized waptagent in batch mode :
- copy **waptagent.exe** and pre-waptupgrade locally (if not `/DeployWaptAgentLocally`, upload to server with https);
- option `/DeployWaptAgentLocally` : copy the newly built **waptagent.exe** and `prefix-waptupgrade_xxx.wapt` to local server WAPT repository directory `.\waptserver\repository\wapt\`;
- [NEW] `wapt-get register` : added options for easy configuration of wapt when registering :
- `--pin-server-cert` : pin the server certificate. (check that CN of certificate matches hostname of server and repo);
- `--wapt-server-url` : set `wapt_server` setting in `wapt-get.ini`;
- `--wapt-repo-url` : set `repo_url` setting in `wapt-get.ini`. (if not provided, and there is not `repo_url` set in `wapt-get.ini`, extrapolate `repo_url` from `wapt_server url`);
- [NEW] `wapt-get` : added `check-valid-codesigning-cert / CheckPersonalCertificateIsCodeSigning` action;

### Improvements and fixes

- python libraries updates
- **cryptography from 2.3.1 -> cryptography 2.5.0**;
- **pyOpenSSL 18.0.0 -> pyOpenSSL 19.0.0**;
- [FIX] don't reset `host.server_uuid` in server db when host disconnect from websocket. Set `host.server_uuid` in server db when host gets a token;
- [FIX] modify `isAdminLoggedIn` to try to fix cases when we are admin but function return false;
- [FIX] ensure valid package name in package wizard (issue959);
- [FIX] regression when using python cryptography 2.4.2 openssl bindings for windows XP agent (openssl bindings of the python cryptography default WHL `>= 2.5` does not work on Windows XP);
- [FIX] trap exception when creating db tables from scratch fails, allowing upgrade of structure;
- [FIX] reduce the risk of *database is locked* error;
- [FIX] deprecation warning for verifier and signer when checking crl signature;
- [FIX] `persistent_dir` calculation in package's `call_setup_hook` when `package_uuid` is None in local wapt DB (for clients migrated from pre 1.7 wapt, error None has no len() in audit log);
- [FIX] regression don't try to use `host_certificate / key` for client side ssl authentication if they are not accessible;
- [IMP] define proxies for crl download in **wapt-get scan-packages**;
- [IMP] fixed bad normalization action icon;
- [IMP] paste from clipboard action available in most packages editing grid;
- [IMP] propose to define package root dev path, package prefix, waptagent or new private key / certificate when launching waptconsole;
- [IMP] remove the need to define waptdev directory when editing *groups / profiles / wua packages / self-service* packages;

- [IMP] grid columns translations in French ;
- [IMP] waptexit responsiveness improvements. Events check thread and tasks check thread are now separated.
- [NEW] added ClientAuth checkbox when building certificate in waptconsole ;
- [NEW] added `--quiet -q` option to `postconf.py`
- [MISC] add an example of client side cert auth
- [ADD] added clientAuth extended usage to x509 certificates (default True) for https client auth using personal certificate ;
- [NEW] use of ssl client cert and key in waptconsole for server authentication ;
- [FIX] ssl client certificate auth not taken in account for server api and host repository ;
- [ADD] added `is_client_auth` property for certificates ;
- default *None* for `is_client_auth` certificate / CSR build ;
- don't fallback to host's client certificate authentication if it is not clientAuth capable (if so, http error 400) ;
- [MISC] waptcrypto : added SSLPKCS12 to encapsulate pcks#12 key / certificate in certificate store ;
- [MISC] added splitter for log memo in Packages for hosts panel ;
- [FIX] store fixes ;
- [FIX] be tolerant when no `persistent_dir` in *waptwua* packages ;
- min wapt version 1.7.3 for self service packages and *waptwua* packages,
- [FIX] WsusUpdates has no attribute `downloaded` ;

### 81.4.19 WAPT-1.7.3.7 (2019-02-19)

(hash 373f7d92)

#### Bug fixes

- [FIX]] softs normalization dialog closed when typing F key (**Enterprise** only) ;
- [IMP] include waptwua in nginx wapt server windows locations (**Enterprise** only) ;
- [FIX] force option from service or websockets not being taken in account in `install_msi_if_needed` or `install_exe_if_needed` ;
- [IMP] improved win updates reporting (uninstall behavior) (**Enterprise** only) ;
- [ADD] added uninstall action for winupdates in waptconsole (**Enterprise** only) ;
- [FIX] reporting from dmi « size type » fields with non integer content (**Enterprise** only) ;

#### Improvements

- [IMP] waptexit : allow minimize button ;
- [IMP] waptexit : layout changes ;
- [IMP] AD Auth : less restrictive on user name sanity check (**Enterprise** only) ;
- [IMP] handling of updates of data for winupdates with additional download urls (**Enterprise** only) ;
- [ADD] added some additional info fields to WsusUpdates table (**Enterprise** only) ;
- [ADD] added filename to Packages table for reporting and store usage (**Enterprise** only) ;
- [ADD] added uninstall win updates to waptconsole (**Enterprise** only) ;
- [ADD] added windows updates uninstall task capabilities (**Enterprise** only) ;
- [ADD] added filename to Packages table ;
- [IMP] increased default clockskew tolerance for client socket io ;

## 81.4.20 WAPT-1.7.3.5 (2019-02-13)

### Bug fixes

- [FIX] regression in package filenames (missing \_);
- [FIX] mismatch for waptconsole [global] waptwua\_enabled setting;
- [FIX] default waptconsole *EnableWaptWUAFeatures* to True;

## 81.4.21 WAPT-1.7.3.4 (2019-02-13)

### Bug fixes

- [FIX] waptexit : fixed install of and empty list of Windows Updates (**Enterprise** only);
- [FIX] wapt-get.exe WaptWUA commands : fixed import of waptwua client module for waptwua-scan download install (**Enterprise** only);
- [FIX] *install\_delay* for Windows Updates stored as a *time\_delta* in waptadb (**Enterprise** only);

### Improvements

- [ADD] versioning on group packages filenames;
- [ADD] button to create AD Host profiles (package automatically installed/removed based on AD Group memberships)
- [IMP] reduce wapttray notifications occurrences. *notify\_user = 0* per default
- [FIX] waptexit : fixed details panel does not show the pending packages to install;
- [FIX] always install the missing dependencies in install (even if upgrade action should have queued dependencies installs before) for some corner known cases;
- [FIX] get server certificate chain popup action when building the waptagent;
- [ADD] action to create a key / certificate in waptconsole conf;
- [IMP] hide inactive / disabled WaptWUA actions in Host popup menu;
- [ADD] checkbox to display newest only for groups;
- [ADD] waptconsole config parameter *licences\_directory* to specify the location (directory) of licenses (**Enterprise** only);
- [IMP] waptagent build dialog : Removed the *Append host's profiles* option;
- [IMP] remove waptenterprise directory if waptsetup community is deployed over a waptenterprise edition;

## 81.4.22 WAPT-1.7.3.3 (2019-02-11)

- [IMP] Core :
- better support for locales, maturities and architecture packages filtering;
- [NEW] Self service rule packages (**Enterprise** only) :
- Package to define which packages can be installed / remove for groups of users;
- WAPT Windows Updates rules packages (**Enterprise** only);
- [NEW] package to define which Windows Updates are allowed / forbidden to be deployed by Wapt WUA agents;
- **waptagent** build :
- [ADD] option for *use\_fqdn\_as\_uuid* when building waptagent.exe;
- [ADD] option to define the profile package to be deployed upon Wapt install on hosts;
- [ADD] options to enable WaptWUA (Windows updates with Wapt) (**Enterprise** only);
- Host Profile packages (**Enterprise** only) :

- [IMP] specific packages (like Group packages) which are installed or removed depending of wapt-get.ini [global] host\_profiles ini key;
- [NEW] if a *profile* package name matches Computer's AD Groups, it is deployed automatically;
- Reporting (**Enterprise** only) :
- [NEW] import / export queries as json files;
- [IMP] softwares names normalization as a separate dialog ;
- **waptexit** :
- [IMP] reworked to make it more robust;
- [IMP] takes in account packages to remove ;
- [IMP] takes in account Wapt WUA Updates (**Enterprise** only) :
- command line switch : /install\_wua\_updates ;
- wapt-get.ini setting : [waptwua] install\_at\_shutdown = 1 ;
- checkbox in waptexit to skip install of Windows Updates ;
- **waptconsole** Custom commands :
- [NEW] ability to define custom poppmenu commands which are launched for the selection of hosts. Custom variables {uid} ;
- Other improvements :
- [IMP] French translations fixes ;

### 81.4.23 Changelog 1.7.2

- [NEW] Reporting (**Enterprise** only) :
- basic SQL reporting capability ;
- duplicate action / copy paste for reporting queries ;
- [ADD] setuphelpers : added helpers processes\_for\_file and get\_computer\_domain ;

### Libraries updates

- **python 2.7.15** on Windows ;
- **openssl-1.0.2p** ;
- upgraded to **python-requests 2.20.0** (Security Fix) ;

### Improvements

- [IMP] don't refresh GridHostsForPackage if not needed (**Enterprise** only) ;
- [IMP] don't add a newline to log text output for LogOutput ;
- [IMP] improved handling of update\_host\_data hashes to reduce amount of data sent to server on each **update\_server\_status** ;
- [IMP] set python27.dll path in wapt-get and **waptconsole.exe** (fix cases with multiple python installations) ;
- [FIX] removal of packages when upgrading host via websockets ;
- [IMP] don't get host capabilities if not needed when updating ;
- [IMP] don't check package control signatures in wapt-get when loading list of packages for development tasks ;
- [IMP] Moved static waptserver assets to a /static root split base.html and index.html templates for blueprints ;
- [FIX] selective pending wua install or downloads (**Enterprise** only) ;
- [FIX] wua updates filter logic (**Enterprise** only) ;

- [IMP] uninstall host packages if `use_hostpackages` is set to false ;
  - add a forced update in the task loop when host capabilities have been changed ;
  - include `use_host_packages` and `host_profiles` in host's capabilities ;
  - [FIX] regression not removing implicit packages.
  - [IMP] more tolerant to unicode errors in `update_host_data` to avoid hiding actual exception behind an encoding exception.
  - [FIX] order of columns not kept when exporting reports (**Enterprise** only)
  - [IMP] `install_msi_if_needed`, `install_exe_if_needed` : check if `killbefore` is not empty or None
  - [IMP] changed tasks's progress and runstatus to property
  - [FIX] Audit aborted due to exception : "NoneType" object is not iterable (**Enterprise** only)
  - [ADD] setuphelpers : Add `get_app_path` and `get_app_install_location`
  - add `fix_wmi` procedure to re-register WMI on broken machines
  - some wmi fallbacks to avoid unregistered machines when WMI is broken on them
  - [ADD] Online wua scans (**Enterprise** only)
  - [ADD] random `package_uuid` when signing a package metadata which could be used later as a primary key :
  - creates a random `package_uuid` when installing in DEV mode ;
  - creates a random `package_uuid` when installing a package without `package_uuid` ;
  - [IMP] moved and renamed `EnsureWUAUServRunning` to setuphelpers ;
  - [ADD] `pending_reboot_reasons` to inventory ;
  - [IMP] display package version for missing packages ;
  - [ADD] **wapt-get sign-packages** : added setting `maturity` and `inc` version in `sign-packages` action ;
  - [ADD] WindowsUpdates's host History grid below WindowsUpdate grid (**Enterprise** only) ;
  - [IMP] store Host Windows update history in server DB (**Enterprise** only) ;
  - [IMP] keep selected or focused rows in grids ;
  - [IMP] updates Packages table when uploading a Package / Group. This table is meant mainly for reporting purpose ;
  - [IMP] disable indexes for some BinaryJson fields ;
  - [FIX] windows update `install_date` reporting (**Enterprise** only) ;
  - [ADD] checkbox to enable `use_fqdn_as_uuid` when building **waptagent.exe** ;
  - [IMP] change default value for `upgrade_only_if_not_process_running` ;
  - [IMP] changed naming of organizational *unit* packages to remove ambiguity with comma in package name and comma to describe the list of packages depends / conflicts :
- Replace “,” with “\_” when editing package (**Enterprise** only) ;
- [ADD] `waptexit` : added priorities and `only_if_not_process_running` command line switches ;
  - [IMP] `waptupgrade` : changed `windows_version` and `Version` ;
  - [ADD] setuphelpers `windows_version` : added `members_count` ;
  - [IMP] `waptutils.Version` : strip members to `members_count` if not *None* ;
  - [ADD] control attributes editor keywords `license homepage package_uuid` to local `waptservice` db ;
  - [ADD] short fingerprint to repr of `SSLCertificate` ;
  - [IMP] be sure password gui is visible even if parent window is not ;
  - [ADD] gui for private key password dialog if `--use-ggui` ;
  - [ADD] `--use-gui` to **wapt-get.exe** command line arg to force use of `waptguihelper` for server credentials when registering ;

### 81.4.24 WAPT-1.6.2.7 (2018-10-02)

This is a bugfix release for 1.6.2.5 :

- [FIX] *waptexit* : changed the default value of *upgrade\_only\_if\_not\_process\_running* parameter to *False* instead of *True* : if *upgrade\_only\_if\_not\_process\_running* is *True*, the install tasks for packages with running processes (*impacted\_process*) are skip-ped;

if *upgrade\_only\_if\_not\_process\_running* is *False*, the install tasks for packages with running processes may impact the user if the installer kills the running processes;

- [FIX] *waptwua* : take in account Windows Updates *RevisionNumber* attribute to identify uniquely an Update in addition to *UpdateID* field (**Enterprise** only). This fixes the 404 error when downloading missing windows updates on a client.

### 81.4.25 WAPT-1.6.2.6 (2018-09-26)

This is a bugfix release for 1.6.2.5 :

- [FIX] WAPTServer Enterprise on Windows : added proper upgrade path from **PostgreSQL 9.4** (used in WAPT 1.5) to **PostgreSQL 9.6** which is required for WAPT-Windows Update :
- new database binary and data directory path are suffixed with -9.6;
- old data is suffixed with -old after migration;
- [FIX] upgrade script for **MongoDB** upgrade (WAPT 1.3) to **PostgreSQL** used since WAPT 1.5;
- [FIX] regression on WMI / DMI inventory which may be not properly sent back to the server;

### 81.4.26 WAPT-1.6.2.5 (2018-09-14)

[NEW] Main new features if you are coming from 1.5 :

- per package *Audit* feature (**Enterprise** only);
- *WAPT managed Windows Updates* tech preview (**Enterprise** only);
- wizards to guide post configuration of Windows server and first use of **waptconsole**;
- **waptconsole**/ private repo page : added a grid which shows the computers where the selected package is installed;

It includes numerous changes over the 1.5.1.26 version.

#### New

- [NEW] per package audit feature :
- def *audit()* hook function to add into package's *setup.py*. By default, check *uninstall key* presence in registry :
- **wapt-get audit**;
- **wapt-get -S audit**;
- **wapt-get audit <packagename>**;
- right click in *waptconsole* on machines or installed packages/ Audit package;
- synthetic audit status for each machine;
- for each installed package : *last\_audit\_status*, *last\_audit\_on*, *last\_audit\_output*, *next\_audit\_on*;
- scheduled globally with *wapt-get.ini* parameter [*global*] :

waptaudit\_task\_period = 4h or in package's control file :

audit\_schedule = 1d

- audit log displayed in **waptconsole** below installed package grid if *Audit Status* column is focused;
- [UPD] updated python modules
- [IMP] build with **Lazarus 1.8.2** instead of **CodeTyphon 2.8** for the Windows executables ;
- better strings encoding handling and easier to setup for the development ;

### Known issues

- **PostgreSQL 9.6** is required for WAPT WUA tech preview (Debian Jessie not supported);
- WAPT 1.6 includes one more security layer in the agent to server connection. After server upgrade, the client desktops won't be able to connect to the server as long as they have not been upgraded themselves. If you require to be able to remotely manage the WAPT agent while the agent has not yet been upgraded, it is necessary to set *allow\_unauthenticated\_connect* to *True* in `waptserver.ini` ;

### Fixes

- [FIX] add AD Groups as Hosts dependencies in **waptconsole** ;
- [FIX] remove image on reachable column if no status has been sent yet ;
- [FIX] Organizational Units WAPT packages not being installed when there are spaces in DN ;
- [FIX] Operational error when host are trying to reconnect but are not registered ;
- [FIX] fill in *created\_on* db fields on win updates data ;
- [IMP] debian server postinst : remove old pyc files ;

### Changes

- [IMP] Improved WAPT console setup Wizard ;
- [ADD] *allow\_unauthenticated\_connect* defaults to *allow\_unauthenticated\_registration* if it is not explicitly set in `waptserver.ini` file (This will ease migration from 1.5 to 1.6) ;
- [IMP] Escape key on password edit of login moves focus to configuration combo ;
- [IMP] PackageEntry.asrequirement() : removed space between package name and version specification ;
- [IMP] missing *install\_date* in *insert\_many* for some updates ;
- [ADD] add force arg for WAPTUpdateServerStatus action ;
- [IMP] don't includes `setup.py` in initial host's packages inventory, and full inventory ;
- [IMP] allow to use installed **waptdeploy.exe** without retry/ignore dialog ;
- [IMP] be sure error is reported properly in **socketio** ;
- [IMP] added *package\_uuid* and homepage package attributes ;
- [IMP] added installed on columns for host wsus updates ;
- [FIX] WUA grid layout saving ;

## 81.4.27 WAPT-1.6.2.2 (2018-07-16)

### Known issues

- **PostgreSQL 9.6** is required for WAPT WUA tech preview (Debian Jessie not supported);
- the authentication of client connections to the WAPT websockets server is not compatible with pre-1.6.2 wapt clients. During migration, if you want to keep the connection with clients, you have to disable the authentication with the parameter : *allow\_unauthenticated\_connect* = 0 in server's configuration file `waptserver.ini`. When all clients have migrated, this can be removed ;



## New

- [NEW] wizard for the initial configuration of **waptserver** on Windows ;
- [ADD] wizard for the initial configuration of **waptconsole** connection parameters ;
- [ADD] **Enterprise only** : waptconsole/ private repo page : added a grid which shows the computers where the selected package is installed ;
- [NEW] **Enterprise only** : WAPT WUA Windows Updates management technical preview :
  - activate with `waptwua_enabled = 1` in `wapt-get.ini` file on the client ;
  - scan of updates on Windows clients with the IUpdateSearcher Windows API and the `wsuscan2` cab file from Microsoft ;
  - additional page in *WAPTconsole* host inventory for Windows updates status reported (HostWsus model) ;
  - additional page in *WAPTconsole* for the consolidated view of all updates reported by hosts (WsusUpdates model) ;
  - periodic task on server to check and download newer version of `wsuscan2` cab file from Microsoft (daemon/ service wapt-tasks) ;
  - periodic Task on server to download missing windows updates files as reported by Windows client after scan :
  - missing files are downloaded if one of the client should install it and has not yet a copy in its local windows update cache ;
  - downloads are logged in *WsusDownloadTasks* model ;

## Changes

- [ADD] field in hosts table to keep the hashes of sent host data, so that clients can send only what needs to be updated ;
- [ADD] `db_port` server config parameter if **postgres** server is not running on standard port 5432 ;
- [ADD] editor optional attribute for package control, used in `register_windows_uninstall` helper if supplied ;
- [IMP] websocket authentication with a timestamped token obtained from server with client SSL certificate on server with client SSL certificate ;
- [IMP] json responses from **waptserver** are gzipped ;

## Fixes

- [IMP] forced host uuid ;
- [IMP] forced computer AD Organizational unit ;
- [IMP] public certs dir ;
- [FIX] caching of negative result for certs chain validation ;
- [IMP] refactoring of server python modules (*config, utils, auth, app, common, decorators, model, server*) for the enterprise modularity ;
- [FIX] timezone file timestamp handling for http download ;

## Python modules updates

- upgrade to **peewee 3.4** ;
- upgrade to **eventlet==0.23.0** ;
- upgrade to **huey 1.9.1** ;
- **eventlet 0.20.1 -> eventlet 0.22.1** ;

### 0.22.1 :

- [IMP] event : `Event.wait()` `timeout=None` argument to be compatible with upstream CPython ;
- [IMP] greendns : Treat `/etc/hosts` entries case-insensitive. Thanks to Ralf Haferkamp ;

### 0.22.0 :

- [IMP] dns : reading `/etc/hosts` raised `DeprecationWarning` for universal lines on Python 3.4+. Thanks to Chris Kerr ;
- [IMP] green.openssl : Drop `OpenSSL.rand` support. Thanks to Haikel Guemar ;

- [IMP] green.subprocess : keep CalledProcessError identity. Thanks to [Linbing@github](#);
- [IMP] greendns : be explicit about expecting bytes from sock.recv. Thanks to Matt Bennett;
- [IMP] greendns : early socket.timeout was breaking IO retry loops;
- [IMP] GreenSocket.accept does not notify\_open. Thanks to orishoshan;
- [IMP] patcher : set locked RLocks' owner only when patching existing locks. Thanks to Quan Tian;
- [IMP] patcher : workaround for monotonic « no suitable implementation ». Thanks to Geoffrey Thomas;
- [IMP] queue : empty except was catching too much;
- [IMP] socket : context manager support. Thanks to Miguel Grinberg;
- [IMP] support : update **monotonic 1.3** (5c0322dc559bf);
- [IMP] support : upgrade bundled to **dnspython 1.16.0** (22e9de1d7957e) <https://github.com/eventlet/eventlet/issues/427>;
- [FIX] websocket leak when client did not close connection properly. Thanks to Konstantin Enchant;
- [IMP] websocket : support permessage-deflate extension. Thanks to Costas Christofi and Peter Kovary;
- [IMP] wsgi : close idle connections (also applies to websockets);
- [IMP] wsgi : deprecated options are one step closer to removal;
- [IMP] wsgi : handle remote connection resets. Thanks to Stefan Nica;

### 0.21.0

- [IMP] new timeout error API : `.is_timeout=True` on exception object. It's now easy to test if network error is transient and retry is appropriate. Please spread the word and invite other libraries to support this interface;
- [IMP] hubs : use monotonic clock by default (bundled package); Thanks to Roman Podoliaka and Victor Stinner
- [IMP] dns : EVENTLET\_NO\_GREENDNS option is back, green is still default;
- [IMP] dns : hosts file was consulted after nameservers;
- [IMP] wsgi : `log_output=False` was not disabling startup and accepted messages;
- [IMP] greenio : Fixed OSError : [WinError 10038] Socket operation on nonsocket;
- [IMP] dns : EAI\_NODATA was removed from RFC3493 and FreeBSD;
- [IMP] green.select : fix `mark_as_closed()` wrong number of args;
- [NEW] added zipkin tracing to eventlet;
- [IMP] db\_pool : proxy `Connection.set_isolation_level()`;
- **Flask-socketio 2.9.2** -> **Flask-socketio 3.0.1**;
- **python-engineio 2.0.1** -> **python-engineio 2.0.4**;
- **python-socketio 1.8.3** -> **python-socketio 1.9.0**;
- upgrade to **websocket-client 0.47**;

## 81.4.28 WAPT-1.6.2.1 (2018-07-04)

### New features

- [ADD] `def audit()` optional hook in package is called periodically to check compliance. Log and status is reported in server DB and displayed in console (**Enterprise**).
- [ADD] WSUS tech preview : based on local Windows update engine and WSUSSCAN2 cab Microsoft file. WAPT server act as a caching proxy for updates. Scanning for, downloading and applying Windows updates can be triggered from console on workstations (**Enterprise**). A new `waptasks` process is launched on the server to download updates and `wsusscan` cab from Internet.

## Changes / Improvements

- [IMP] better utf8 handling ;
- [IMP] **wapt-get make-template** from a directory creates a basic installer for portable apps ;
- [IMP] wapt-get, waptexit : Removed ZeroMQ message queue on the client, replaced by simple http long polling to monitor tasks status ;
- [IMP] waptconsole : Replaced blocking timer based http polling for tasks status by threaded http long polling ;
- [IMP] waptconsole : Filter hosts on whether current personal certificate signature is authorized for remote tasks (**Enterprise**). If same server is used for several organizations, it allows to focus on own machines. This supposes that different CA certificates are deployed depending on the client host's organization. In this release, the filtering is not enforced and not cryptographically authenticated ;
- [CHANGE] renamed waptservice.py to service.py and waptserver.py to server.py, activated absolute import for all python sourced absolute import for all python sources ;
- [REMOVED] *use\_http\_proxy\_for\_template* parameter (setting is now in [wapt-templates] repo) ;

### waptservice

- [ADD] handling of WUA tasks (Scan, download, apply updates) (**Enterprise**) ;
- [ADD] handling of auditing tasks ;

### waptserver

- [ADD] tasks queue (**Huey**) for the WSUS background tasks (**Enterprise**) ;
- [IMP] gzip compression activated on the **nginx** configuration ;

### wapttray

- [ADD] option in wapt-get.ini to hide some items :
- *hidden\_wapttray\_actions* : comma separated list of :

*LaunchWAPTConsole, register, serviceenable, reloadconfig, cancelrunningtask, cancelalltasks, showtasks, sessionsetup, forceregister, localinfo, configure* ;

- [CHANGE] use long polling instead of **zmq** ;
- [IMP] stop/ start/ query waptservice using a thread to avoid gui freeze ;

### Fixes

- [FIX] waptguihelper : be sure to load the proper python27.dll ;
- [FIX] core : forward *force* argument from console to *setup.py* install() hook ;
- [FIX] overwrite psproj package file when editing a package to fix path to WAPT python virtualenv and add new debug actions ;

### Modules updates

- [UPD] GUI Binaries are built with **Lazarus 1.8.2 / fpc 3.0.4** instead of **CodeTyphon 2.8**;
- [UPD] **peewee 3.0.4**;
- [UPD] **eventlet 0.23.0**;
- [UPD] **huey 1.9.1**;
- [UPD] **pywin32** rev 223;
- [UPD] **Flask-socketio 2.9.6**;
- [UPD] **engineio.socket 2.0.4**;
- [UPD] **websocket-client 0.47**;
- [UPD] **pyOpenSSL 17.5.0**;
- [UPD] **request 2.19.1**;

### Known issues

- *unit* type of packages (with AD DN style names) are not well handled by local WAPT self service, because of commas in name.

## 81.4.29 WAPT-1.6.1.0 (2018-06-21)

### Fixes

- [FIX] av potential cause in wapttray;
- [IMP] buffer LogOutput;
- [FIX] wait task result loop in waptserver;
- [FIX] bad acl on waptservice;
- [FIX] repo timeout not taken in account;
- [FIX] bad parameter for `repo_url` and `[wapt-host]` section;
- [FIX] waptexit AV potential cause;
- [FIX] make `isAdmin` non blocking as a workaround for false positive checks;
- [FIX] use timeout parameter when importing external package;
- [FIX] pass timeout parameter when importing;
- [FIX] bad `repo_url` config naming;
- [FIX] calc hash when compiling if file does not exist;
- [FIX] repo timeout is float;
- [FIX] custom zip corruption when signing a package with non ascii filenames;
- [FIX] check `wapt_db` is assigned when rollbacking;
- [IMP] logging in events;
- [FIX] installed packages section is incorrectly reported as *base* instead of *unit* or *host* in waptconsole;
- [IMP] ensure manual service wua is running when using command line;
- [UPG] Python modules updates :
- upgrade to **peewee 3.4**;
- upgrade to **eventlet==0.23.0**;
- upgrade to **huey 1.9.1**;
- [CHANGE] replace `eventprintinfo` with `LogOutput`;
- [ADD] `waptwua_enabled` config parameter;
- [IMP] missing `ensure_list` `waptwua_enabled` config parameter;
- [IMP] default `waptwua_enabled` to `None` to avoid wuauserv service configuration change;
- [ADD] missing columns for window updates;
- [ADD] action in waptconsole to show help on KB;

- [IMP] waptray cosmetic : hide duplicated separators in tray popup menu when some actions are hidden ;
- [ADD] http\_proxy ini setting for the server external download operations ;
- [IMP] waptray : Start and stop WAPTservice using a thread to avoid gui freeze ;
- [IMP] Pure FPC PBKDF2 password hash calc for postconf ;
- [IMP] refactor server code to share app and socketio instances ;
- [FIX] forward the « force » argument (command line and through the websockets) to the install() setup.py hook ;
- [FIX] do not display all missed events at tray startup in waptray ;
- [FIX] no default audit\_period ;
- [REMOVED] **zeromq**, replaced by long http polling between waptray, wapt-get and waptservice ;

### 81.4.30 WAPT 1.5.1.26 (2018-07-12)

#### Bug fixes

- [IMP] revert monkey\_patch for server on windows. No reason to exclude thread ;
- [ADD] allow\_unauthenticated\_connect server config (default *false*) ;
- [FIX] CRITICAL update\_host failed UnboundLocalError(« local variable “result” referenced before assignment ».);
- [FIX] <https://roundup.tranquil.it/wapt/issue951> ;
- [FIX] <https://forum.tranquil.it/viewtopic.php?f=13&t=1160ix> ;
- [FIX] <https://forum.tranquil.it/viewtopic.php?f=13&t=1160> ;
- [FIX] `init_workdir.bat` ;
- [FIX] returns a token when updating host data for websocket authentication ;
- [IMP] rewrite package psproj when editing (to fix wapt basedir paths) ;
- [FIX] %s -> %d format string for expiration warning message ;
- [FIX] host\_certificate not found for waptstarter ;
- [ADD] some dev build scripts ;

### 81.4.31 WAPT-1.5.1.24 (2018-07-04)

#### Bug fixes

- [FIX] zipfile python library bug for packages which contains files with non-ascii filenames. Signed WAPT packages were corrupted in this case ;
- [FIX] deadlocks on server database when simultaneous DB connections is larger than 100 (default maximum connections configured by default on postgresql) ;
- [FIX] waptconsole crash on warning message when license is about to expire (**Enterprise** only) ;
- [FIX] %s -> %d format string for expiration warning message ;
- [FIX] host\_certificate not found for waptstarter ;
- [FIX] waptserversetup.iss to include enterprise modules (**Enterprise**) ;
- [FIX] download link to waptsetup and waptdeploy on server index page for Windows ;

### Modules updates

- **requests 2.19.1**;
- **Rocket 1.2.8** - Don't try to resurrect connections that timeout. Increase the timeout ... to decrease the likelihood ;
- handle PyPi only supports HTTPS/TLS downloads now ;
- fix the problem that when body is empty no terminating ; chunk is sent for chunked encoding.
- avoid sending the terminating chunk in case it's a HEAD request ;
- fix the problem that when body is empty no terminating chunk is sent for chunked encoding ;
- explicitly set the log level to warning ;
- fix bug « Threadpool grows by negative amount when max\_threads = 0 » ;
- don't try to resurrect connections that timeout. Increase the timeout to decrease the likelihood ;

### 81.4.32 WAPT-1.5.1.23 (2018-03-28)

#### Changes

- [IMP] waptexit : display a custom PNG logo if one is created in %WAPT\_HOME%\templates\waptexit-logo.png ;
- [IMP] nssm.exe is signed with Tranquil IT code signing key ;
- waptconsole : Add locale and maturity columns in packages status grid ;
- waptconsole : wapagent wizard ; be sure to get a relative path when checking cert validity ;
- waptsetup : Add /CopyPackagesTrustedCA and /CopyServersTrustedCA command line parameters to allow deployment of wapt with specific certificates with GPO for wapt without recompiling waptsetup ;

Example :

```
C:\tmpwaptdeploy --hash=e17c4eddd45d34000df0cfe64af594438b0c3e1ee9791812516f116d4f4b9fa9
--minversion=1.5.1.23 --waptsetupurl=http://buildbot/~tisadmin/wapt/latest/waptsetup.exe
--setupargs=/CopyPackagesTrustedCA=c:\tmptranquilit.crt --setupargs=/CopyServersTrustedCA=c:\tmpsrvwapt.
mydomain.lan.crt --setupargs=/verify_cert=sslserverwapt.mydomain.lan.crt --setupargs=/
repo_url=https://srvwapt.mydomain.lan/wapt --setupargs=/waptserver=https://srvwapt.mydomain.lan
--setupargs=/DIR=c:wapt
```

#### Bug fixes

- [FIX] waptconsole : regression introduced in 1.5.1.22. Unable to login if server has not a FQDN ;
- [FIX] setuphelpers : winstartup\_info fallback when COMMON\_STARTUP folder does not exist, preventing a client to register properly ;
- [FIX] version/ revision in wapttray display the git hash instead of old svn revision number ;
- [FIX] waptconsole : update French translation for certs bundle hint ;
- [FIX] waptconsole : compare properly packages when number of version members differs 1.3 -<> 1.3.1 for example ;

### 81.4.33 WAPT-1.5.1.22 (2018-03-27)

#### Bug fixes

- [FIX] add Active Directory groups;
- [FIX] newest only with locale, architecture and maturity;
- [FIX] Import from external repository with mixed locale, architecture and maturity;
- [ADD] `--setupargs` to **waptdeploy**;
- [FIX] RPM;
- [FIX] Enterprise build (**Enterprise** only);
- [IMP] different icons for WAPT Community and Enterprise editions;
- [IMP] switch to Community features when no licence instead of aborting (**Enterprise**);
- some up to date Installed Packages marked as upgradable because of bad comparison `maturity None/ maturity`;
- [IMP] `depends` and `conflicts` fields of `HostsPackagesStatus` table limited to 800 chars -> type changed to `ArrayField` to handle unlimited number of dependencies;
- [NEW] git python module added as part of WAPT libraries;
- [IMP] list organizational *unit* packages in group package table (**Enterprise**);
- [FIX] MongoDB to PostgreSQL database upgrade script;
- [FIX] licence/ hosts count/ expiry check (**Enterprise**);
- [FIX] relative path for `verify_cert`;

#### Known issues

- When waptserver is searched with DNS SRV query (`dnsdomain param`), kerberos register authentication is not working.

### 81.4.34 WAPT-1.5.1.21 (2018-03-13)

#### Global architecture

- [IMP] multiple languages for description of packages. English, French, German, Spanish, Polish are handled as a start point. More to be added in the future;
- [IMP] the description columns in waptconsole displays either languages depending on `language` setting in `waptconsole.ini`. In packages, `description_fr`, `description_en`, etc... have been added;
- [IMP] when renaming hosts, old host package (matching previous host uuid) is now « removed » instead of forgotten;
- [NEW] Handle AD organizational unit packages (**Enterprise** only);
- [NEW] package attributes :
- `locale` attribute : A computer can be configured to accept only packages with a specific locale;
- `maturity` attribute : stores status like *DEV*, *PREPROD*, *PROD* to describe the level of completion of the package. Computers can be configured to accept packages with specified maturities. Default packages maturity of computer is both the empty one and *PROD*;
- `impacted_process` attribute : csv list of process names which would be killed before install (**install\_msi\_if\_needed**, **install\_exe\_if\_needed**) and uninstall (by the mean of `uninstallkey` list). Could be used too in the future for « soft » upgrade remote action which upgrade softwares while they are not running;

### Setup/ WAPT upgrades

WAPTuupgrade package :

- [IMP] increased lifetime for upgrade task windows scheduler trigger for computers which are down for many days when upgrading;
- [ADD] trigger at start of the computer;

### WAPTconsole

- [IMP] display of the list of embedded trusted packages certificates when building the custom waptagent installer;

### Bug fixes

- [FIX] handle unicode filepaths for Packages Wizard;
- [IMP] work in progress improvement of unicode handling globally in WAPTconsole;
- [FIX] use proxy if needed for « download and edit » from external repo;

### Setuphelpers

- [FIX] bug in `create_programs_menu_shortcut` and `create_user_programs_menu_shortcut`. Shortcuts were created in `startup` and not `startup/programs`.

## 81.4.35 WAPT-1.5.1.19 rc1 (2018-03-08)

### Global architecture

There is now some additional support for packages localization.

In Package control file, the `description_fr`, `description_en`, `description_de`, `description_pl`, `description_es` can be used to give description in respective french, english, german, polish languages.

If not set, the base description is used.

### WAPTconsole

## 81.4.36 WAPT-1.5.1.18 rc1 (2018-02-27)

### Global architecture

There is a significant internal change on how python libraries are managed inside WAPT. This has implications on the way python scripts are launched. This change is only relevant for peoples launching WAPT processes manually.

We have removed the (not clean) `sys.path` manipulations inside wapt python scripts sources. The consequence is that all python scripts must be run with prior setting `PYTHONHOME` and `PYTHONPATH` pointing to WAPT home directory (`/opt/wapt` on Linux).

Failing to do so results in scripts claiming that libraries are missing.

On Linux waptserver, libs are now in the default `/opt/wapt/lib/python2.7` location instead of using non standard former one.

- [IMP] WAPT has its own full python environment for libraries, even when debugging. Before, system wide python27 installation was needed for **PyScripter** to run.



Now, **PyScripter** can be started with a special batch file `waptpyscripter.bat` which sets the environment variables for python (PYTHONHOME and PYTHONPATH) and run **PyScripter** with python dll path set to wapt own copy.

- [NEW] Command line scripts with proper environment :
- `wapt-serverpostconf` on Linux server to start server postconf.py
- `wapt-scanpackages`
- `wapt-signpackages`
- [NEW] debugging commandline tools which setup python environment properly before running the python script.py before running the python script :
- to debug waptservice, launch in cmd as admin : `runwaptservice.bat` ;
- to debug waptserver, launch in cmd : `runwaptserver.bat` or under linux : `runwaptserver.sh` ;
- to launch **PyScripter** without the need for local system wide python27 install, run `waptpyscripter.bat` ;

## WAPT client

- [IMP] Add local wapt-get.ini settings `packages_whitelist` and `packages_blacklist` to restrict accepted packages from repository based on their package's name ;
  - [IMP] More detailed reporting off host's repositories configuration (now includes dnsdomain, proxy, and list of trusted certificates) ;
  - [FIX] fixed display in the Windows task bar of the login window (to allow in particular the autofill of the password by password managers) ; waptagent failing to compile if keys/ certificates already exist but the certificate had been removed from `C:\wapt\ssl` ;
  - [NEW] Handle AD organizational unit packages (Enterprise edition)
  - [IMP] Fallback to basic auth when a host is registering on waptserver if kerberos is enabled but authentication fails.
  - [IMP] for `wapt-get.exe`, allow to designate configuration `wapt-get.ini` file with `-config` option with base name of user waptconsole ini file (without ini extension) instead of full path. Handy when switching between several configurations. Same behavior as for waptconsole. Example :
- ```
wapt-get -c site3 build-upload c:\waptdev\test-7zip-wapt;
```
- [FIX] Be sure to not loop for ever in websockets retry loop if something is wrong in host waptserver or websocket configuration.
  - [FIX] Update PyScripter project template to use project directory as parameter for debug actions, and use relative paths for filenames.
  - [FIX] incorrect package version comparison. Return True when comparing 1.2-1 to 1.2.1-3 (note : this is not homogeneous with the Version() class behavior. todo : merge both) ;
  - [FIX] waptsetup : register and update must be launched with elevated privileges. So remove `runasoriginaluser` option.
  - [NEW] Introduced attributes `target_os` and `impacted_process` for package's `control` file. They are not yet taken in account.
  - [NEW] Introduced machinery to handle X509 client certificates authentication for repositories and waptserver (specially for public servers) ;
  - [NEW] Introduced classes to generate X509 CRL ;

## Setuphelpers

- [UPD] `setuphelpers.removetree` : Try to remove readonly flag when `remove_tree` reach a Access Denied error ;
- [FIX] unicode handling in shell startup shortcuts ;
- [IMP] `waptutils.wget` can check sha1 or sh256 hashes in addition to md5, and can cache and resume partial downloads ;

### WAPT Console

- [NEW] action in WAPTconsole to plan in near future a restart of waptservice on selected hosts ;
- [IMP] mass host update/upgrade in waptconsole actions are now launched in single shot instead of one host at a time ;
- [NEW] allow to force a `host_dn` in `wapt-get.ini` when host is not in a domain (**Enterprise** only) ;
- [NEW] added timeout parameter for setuphelpers `service_start`, `service_stop` and `service_restart` ;
- [IMP] `group filter list` box is now editable, and one can type a partial group match and press enter to filter on all matching groups. Separator is comma (,). Handle \* at the end of search to find all occurrences even if one group matches exactly ;

### WAPT Server

- [ADD] bat script `migrate-hosts.bat` to set environment for `migrate-hosts.py` ;
- [ADD] `trigger_action.py` script to trigger action on pre 1.5 hosts with reachable 8088 waptservice port from 1.5 server ;
- [FIX] `registration_auth_user` reset to None when reusing host certificate for re-register ;
- [IMP] removed unnecessary dependencies `krb5-user`, `msktutil`, `python-psutil` for waptserver package ;
- [IMP] increase `client_max_body_size` for http post on nginx for large update/ upgrade trigger ;
- fix `signature_clockskew` waptserver config parameter not taken in account ;
- unified loggers for server ;
- have waptserver ask wapt clients to update status using websockets if websocket connection is up but database is not aware of given SID (case where waptserver is restarted but **nginx** is kept up, and restart of waptserver service is fast enough to not trigger a reconnection of the clients) ;
- [FIX] disable proxy for `migrate-hosts` ;

### Known issues

- waptservice : if a system account level http proxy is defined in registry on the windows host, websocket client library tries to use it and fails to connect to the server. Workaround : make an exception for waptserver ;
- waptconsole : if a http proxy is defined in `waptconsole.ini`, section `[global]`, key `http_proxy`, it is used by the waptconsole even if setting `use_proxy_for_xxx` is False Workround : set `http_proxy` to an empty string in `waptconsole.ini` ;
- when using a not self-signed personal certificate, depending of th issuer, the certificate file `<private_dir>mine_cert.crt` can contain the full chain (own certificate, intermediate CA, and root CA). When waptconsole asks if the certificate should be put in authorized client certificate directory (`<wapt-dir>ssl`), the full `crt` file is copied as this. This means that all certificates in `crt` file are authorized, and not only the personal one. This is perhaps not desired ;  
Workaround : check if the personal pem encoded `crt` file contains the full certificates chain. If this is the case, copy in `<wapt-dir>ssl` only the parts of the PEM file matching the certificates you want to trust ;
- SNI is not properly handled by waptconsole code, leading to incorrect error about certificate validation on https server with virtual hosts ;
- Certificates CSR updates (periodical signature, ...) must be managed manually using tools like `easy-rsa`. Only CSR accessible by a URL are supported ;
- proxies are not supported on the server, so CRL can not be updated properly (as far as Distribution Point is defined in certificates) if the server has no direct http access to the distribution points ;
- https certificates are verified on the clients using the bundle defined by the `verify_cert` ini settings. If this setting is simply `True`, the bundle supplied with python libraries is used to check issuers. This bundle is not updated unless WAPT is upgraded, so new issuers or no more trusted issuers are taken in account only at this point. So it is better to deploy your own CA bundle along with wapt and define the `verify_cert` path.
- for 1.5.1.18 rc1, on the linux server, there are broken symbolic links in `lib/python2.7` folder. Next RC does not exhibit this problem ;

### 81.4.37 WAPT-1.5.1.14 (2018-01-09)

- [NEW] Historize in *wapt\_localstatus* PostgreSQL table the dependencies and conflicts of installed packages (to provide an easy way to warn when conflicting package will be installed or should be removed);
- [FIX] load full certificate chain from host packages to check `control` (as it is the case for other types of packages);
- [SEC] regression : check host package control signature right after downloading (it is checked too when starting install);
- [FIX] regression : don't install host package if version is lower than installed one;
- [FIX] don't raise an exception during session-setup if package has no `setup.py`;

#### WAPT Client

- [FIX] intermediate CA pinning : Allow to deploy intermediate CA as authorized package CA without root CA (segregation of rules between entities);
- [FIX] old style print statement (without parentheses) raising an error in *setup-session* or *uninstall* **setup.py** functions;

#### setuphelpers/ libraries

- [IMP] Add *cache\_dir* parameter to **wget** function;
- [IMP] renamed *cabundle* parameter to *trusted\_bundle*;
- [NEW] Add python methods to create certificate from CSR;

#### WAPT Console

- [ADD] checkbox in create waptagent to sign with sha1 in addition to sha256 for old wapt client upgrades;
- [IMP] force host package version to be at least equal to already installed host package (when host package is deleted, version was starting again at 0);
- [FIX] regression : check existing host package signature before editing it;

#### WAPT Server

- [FIX] Force waptserver DB structure upgrade at each server startup;
- [ADD] *db\_connect\_timeout* parameter for pool of waptserver DB connections;
- [NEW] Store *depends* and *conflicts* attributes in waptserver *HostPackagesStatus* PostgreSQL table;

#### Known issues

- SNI is not properly handled by waptconsole code, leading to incorrect error about certificate validation on https server with virtual hosts;
- certificates CSR updates (periodical signature, ...) must be managed manually using tools like easy-rsa. Only CSR accessible by a URL are supported;

### 81.4.38 WAPT-1.5.1.13 (2018-01-03)

- Quelques fallback pour permettre l'utilisation de la console WAPT sous Wine
- Ebauche architecture plugins dans waptconsole.
- Interface GUI pour entrer les mots de passe dans PyScripter
- Action make-template dans installeur crée un paquet vide
- Inclusion de la chaine de certificats du signataire dans le paquet au lieu du seul certificat final
- IMPROVE : gestion des certificats signés par une autorité intermédiaire pour les actions de la console Wapt
- Ajout option pour spécifier fichier de configuration pour waptconsole.
- [FIX] SNI pour la récupération de la chaine de certificats dans waptconsole.
- [ADD] added actions to launch mass updates/ upgrades, offer updates to the users (WAPT Enterprise);
- F5 rafraîchit la liste des paquets
- Changement à distance de la description de l'ordinateur
- Possibilité de configurer plusieurs instances de serveurs Wapt sur un serveur/ VM.
- chunked http upload pour pouvoir uploader des gros paquets sans passer par du scp.
- Ajout installation forcée d'un paquet sur un poste dans la console.
- Ajout option pour masquer les actions avancées (simplification affichage console)
- CN du Certificat / clé machine sont nommés comme l'UUID.
- Si une ou plusieurs dépendances d'un paquet ne peuvent pas être installées, le paquet parent n'est pas installé et est marqué en erreur.
- Memory leak sur le serveur
- Gestion timezone pour validité de certificats
- [SECURITY] prend tous les fichiers en compte dans la vérification des hashes, pas seulement ceux dans le répertoire racine (régression apparue en 1.5 mais non présente en 1.3)

### 81.4.39 WAPT-1.5.1.5 (2017-11-16)

#### Architecture globale

- [NEW] the host packages are now named with the BIOS *UUID* of the machine instead of the *FQDN* (it is possible to use the *FQDN* as the *UUID* with the parameter *use\_fqdn\_as\_uuid* but it may create duplicates in the console);
- le service **waptservice** écoute sur l'adresse de loopback, port 8088 et non plus sur toutes les interfaces. Cela réduit la surface d'attaque potentielle si un attaquant spoofe l'adresse IP du serveur WAPT;
- le service **waptservice** crée au démarrage une connexion Websockets (Socket.IO) vers le serveur pour permettre à la console de déclencher les Update/ Upgrade / Install/ Remove; On ne pass plus par le port 8088 du service;
- [NEW] the Websocket requests from the WAPT console to the WAPT agents are now signed with the key of the *Administrator*. Before, security relied on source IP restriction and the validation of the Administrator's login/ password;
- la base de données d'inventaire est maintenant une base PostgreSQL en remplacement de MongoDB. Cela facilite le requêtage pour un reporting personnalisé, le langage SQL étant mieux connu des administrateurs système;
- l'affichage dans la console d'un grand nombre de machines a été amélioré. L'affichage de plusieurs milliers de machines n'est plus un problème;
- modifier la configuration d'un grand nombre de machines a été rendu largement plus performant;
- la reprise d'un téléchargement partiel de paquet est maintenant possible (interruption lors de l'arrêt ...);
- les clés privées doivent maintenant obligatoirement être protégées avec un mot de passe;

## Console WAPT

- passage en Websockets ;
- gestion des écrans de haute résolution (ex : écrans 4k) ;
- modernisation des jeux d'icônes dans la console ;
- changement à la volée de la description du poste ;
- option pour changer le mot de passe d'une clé ;

## Format des paquets

- la présence du fichier `setup.py` est optionnelle (plus particulièrement, il n'est pas nécessaire pour les paquets groupes et machines qui ne contiennent que des dépendances) ;
- [NEW] if the package contains a `setup.py` file, it **MUST** be signed with a **Code Signing** certificate, otherwise the package **WILL NOT** be installed. The roles are now differentiated between the role of the *Package Deployer* (allowed to sign group and host packages) and the role of *Package Developer* (allowed to sign group, host AND base packages) ;
- lors de la signature du paquet, le certificat du signataire est ajouté dans le paquet (`WAPT/certificate.crt`) ;
- le fichier `manifest` est renommé `manifest.sha256` au lieu de `manifest.sha1` et `signature.sha256` au lieu de `signature` ;
- ajout des attributs suivants au fichier `control` :
  - `signed_attributes` : pour la fiabilité de la vérification ;
  - `min_wapt_version` : le paquet est ignoré (et ne s'installe pas) si wapt n'est pas au moins à cette version ;
  - `installed_size` : le paquet ne s'installe pas s'il n'y a pas au moins cet espace disponible sur le disque système ;
  - `max_os_version` : le paquet est ignoré si Windows a une version supérieure à cet attribut ;
  - `min_os_version` : le paquet est ignoré si Windows a une version inférieure à cet attribut ;
  - `maturity` : *PROD*, *PREPROD*, *TEST* ;
  - `locale` ; *fr*, *en*, etc ;

## Configuration générale des agents

- section explicite `[wapt-host]` pour le dépôt des paquets machines sinon l'url est déduite de `<repo_url>+''-host''` ;
- section explicite `[wapt]` pour le dépôt principal, sinon `<repo_url>` est pris en compte ;
- vérification des certificats activée par défaut pour toutes les connexions https ;
- signature avec du sha256 au lieu de sha1 ;
- prise en compte de paquets signés avec des certificats délivrés par une autorité, déploiement uniquement du certificat de l'autorité ;
- utilisation de l'UUID du client pour le nom des paquets machine au lieu du FQDN ;
- possibilité d'utiliser le FQDN comme UUID au lieu de l'UUID du Bios. (paramètre `use_fqdn_as_uuid`) (ou uuid forcé : paramètre `forced_uuid`) ;
- lorsqu'on signe, on désigne le signataire par son certificat et non sa clé privée. La clé privée est recherchée par wapt dans le même répertoire que le certificat personnel. On incite à avoir un certificat par personne agissant sur WAPT ;
- possibilité de prendre en compte la révocation de certificats (la CSR est fournie aux poste lors de l'update, dans le fichier `Packages`) ;
- re-signature possible sous Linux avec la commande `wapt-signpackage.py` ;
- installation dans Program Files(x86) par défaut ;

### setuptools

- *running\_as\_admin*, *running\_as\_system*;
- correctif sur **add\_shutdown\_script**;
- ajout paramètre *remove\_old\_version* pour **install\_msi\_if\_needed** et **install\_exe\_if\_needed**;

### wapt-get

- ajout fonction **update-package-sources** qui lance la fonction optionnelle **update\_package()** du paquet;
- remplacement de l'option *-private-key* par l'option *-certificate* pour désigner le certificat à utiliser pour signer le paquet. La clé privée est recherchée dans le même répertoire que le certificat;
- remplacement du fichier WAPT/wapt.psproj à chaque édition d'un paquet (pour mettre à jour le chemin vers les modules WAPT suivant l'installation dans C:\wapt ou C:\Program Files (x86)\wapt);
- vérification du certificat serveur lors du **enable-check-certificate** pour éviter de mauvaises configurations;

### wapt-signpackages

- ajout options

```
--if-needed
--message-digest
--scan-packages
--message-digest
```

```
Usage: wapt-signpackages -c crtfile package1 package2
```

Re-sign a list of packages

Options:

```
-h, --help          show this help message and exit
-c PUBLIC_KEY, --certificate=PUBLIC_KEY
Path to the PEM RSA certificate to embed identity in
control. (default: )
-k PRIVATE_KEY, --private-key=PRIVATE_KEY
Path to the PEM RSA private key to sign packages.
(default: )
-l LOGLEVEL, --loglevel=LOGLEVEL
Loglevel (default: warning)
-i, --if-needed     Re-sign package only if needed (default: warning)
-m MD, --message-digest=MD
Message digest type for signatures. (default: sha256)
-s, --scan-packages Rescan packages and update local Packages index after
signing. (default: False)
```

## Console WAPT

- [NEW] all actions sent to the hosts are signed with the Administrator's key ;
- [NEW] generation of a key / certificate pair signed by a Certificate Authority (WAPT Enterprise) ;
- option de créer un certificat **Code Signing** ou non (version Enterprise) ;
- option pour changer le mot de passe d'une clé RSA ;
- option de vérification des certificats lors de la création du **waptagent** ;
- lancement TISHelp (version Enterprise) ;
- limitation du nombre de machines retournées dans la console ;
- ajout filtre *reachable* = poste connecté au serveur WAPT ;
- possibilité de changer la description du poste

## waptserver

- authentification sur une base LDAP (version Enterprise) ;
- utilisation des Websockets pour les actions ;

## waptservice

- le Webservice http de **waptservice** écoute uniquement sur la loopback 127.0.0.1 (donc plus de vérification si port 8088 ouvert sur firewall..) ;
- le **waptservice** se connecte en websocket au serveur WAPT si le paramètre *waptserver* est présent dans *wapt-get.ini* ;
- le paramètre *websockets\_verify\_cert* active la vérification SSL du certificat pour la connexion websockets ;
- affichage de liste des certificats / CA autorisés pour les paquets ;
- affichage signataire paquet ;
- [NEW] *allow\_user\_service\_restart* parameter allows a standard user to restart the WAPT service on her computer ;
- lancement de **tishelp** en mode service par URL /tishelp ;

## Installeur waptagent

- suppression installation **msvcrt** ;
- restent uniquement 2 options : installer le service et lancer *waptray* ;
- options pour une installation silencieuse :
- *dnsdomain* pour la recherche auto wapt et waptserver
- *wapt\_server*
- *repo\_url*
- **waptupgrade** fait systématiquement une installation complète (pas d'installation incrémentale) ;

## Improvements 1.5.0.12-amo -> 1.5.0.16

- *setup.py* pas obligatoire pour uninstall ;
- chemin unicode pour édition de paquets ;
- corrigé la recherche de dépôts en s'appuyant sur les DNS ;
- corrigé \0000 pour PostgreSQL ;
- introduit une option pour avoir une double signature sha1 et sha256 ;
- vérification https pour upload **waptagent** ;
- option *-if-needed* dans **wapt-signpackages** ;
- fix proxy dans import paquets ;
- gestion des révocations de certificats (CSR) ;

- fix attributs requis dans signature actions ;
- *max\_clients* ;
- fix option sans serveur (**waptstarter**) ;
- ajout lancement **tishelp** ;
- force update à l'installation ;

### 81.4.40 WAPT-1.4.0 (2017-05-05)

- pas de release officielle ;
- [NEW] migration sur la base PostgreSQL à la place de MongoDB ;

### 81.4.41 WAPT-1.3.13 (2017-07-25)

#### Security fix

- régression : Package files content check was skipped if signature of **manifest** and **Packages** index file checksum was ok. This regression affects all 1.3.12 releases, but not WAPT <= 1.3.9 and >= upcoming 1.5. In order to exploit this bug, one would need to tamper the **Packages** files either through a MITM (if you do not have valid https certificate check) or a root access on the WAPT server.

#### Other changes

- compatibility with packages signed with upcoming WAPT 1.5. With WAPT 1.5, package are signed with sha256 hashes. An option allows to sign them with sha1 too so that they can be used with WAPT 1.3 without signing them again.
- new package certificate for Tranquil IT packages. previous certificate for package on store.wapt.fr has expired. all packages on store.wapt.fr has been signed again with new key / certificate with both sha1 and sha256 hashes, and WAPT 1.5 signature style (control data is signed as well as files)
- fix for local GPO `add_shutdown_script()` function (thanks jf-guillou !)
- fix for **waptsetup.exe** postinstall actions (**update** / **register**) when running **waptsetup.exe** installer without elevated privileges : added *runascurrentuser* flag
- remove needless python libraries to make install package slimmer

### 81.4.42 WAPT 1.3.12.13 (2017-06-26)

#### Console WAPT

- [NEW] Assistant de création de paquets à partir d'un fichier MSI ou d'un Exe ;
- [NEW] Option dans le menu *Outils* ou par drag drop dans l'onglet dépôt privé ;
- [NEW] Découverte des options silencieuses ;
- [NEW] Utilisation des fonctions **install\_exe\_if\_needed** et **install\_msi\_if\_needed** au lieu d'un simple **run()** pour les exes et les MSI (plusieurs templates de `setup.py` dans `C:\wapt\templates`) ;
- [NEW] Amélioration significative de la vitesse de modification en masse des paquets machines ;
- [NEW] Vérification optionnelle de la signature des paquets que l'on importe d'un dépôt extérieur. La liste des certificats autorisés se trouve par défaut dans `%APPDATA%\waptconsole\ssl` et peut-être précisée dans les paramètres de la **waptconsole**. Le paramètre ini se nomme *authorized\_certs\_dir*. Sinon, les certificats autorisés sont ceux dans `C:\wapt\ssl` ;
- [NEW] Vérification optionnelle du certificat https pour les dépôts extérieurs dans la console ;
- [NEW] Vérification de la signature des paquets machines, groupes et logiciels avant leur modification dans la console ou dans **PyScripter** ;



- [NEW] Lors de l'import d'un dépôt extérieur, possibilité d'éditer le paquet pour inspection plutôt que de le charger directement sur le dépôt de production ;
- [NEW] Changement des URL relatives à la documentation. <https://www.wapt.fr/en/doc/> ;
- [NEW] Possibilité d'actualiser le certificat sans recréer la paire de clés RSA (en particulier pour préciser un Common Name correct, qui apparaît comme le signataire des paquets) ;
- [NEW] HTTPS par défaut pour les URL de dépôt.

### Autres correctifs

- [FIX] Paramètre `AppNoConsole :1` pour NSSM (**waptservice / waptserver**) pour permettre le fonctionnement sur Windows 10 Creators Updates ;
- [FIX] Problème de fichier Zip qui restent verrouillés si une erreur est déclenchée ;
- [FIX] Suppression répertoire temporaire lors de l'annulation d'édition d'un groupe ;
- [FIX] Gestion espace dans les fichiers de projet PyScripter ;
- [FIX] Gestion utf8 / unicode pour certaines fonctions ;
- [FIX] Fix gestion encoding quand `run_not_fatal()` renvoie une erreur ;
- [FIX] remplacement librairie mongo.bson par json natif de python ,
- [FIX] bug dans la synchro des groupes AD avec les paquets WAPT ;
- [FIX] bug « La clé privée n'existe pas » la première fois qu'elle est renseignée si on ne redémarre pas la console ;
- [FIX] bug « redémarrage service wapt » (merci à QGull) ;
- [FIX] possibilité d'avoir des majuscules dans les noms de paquet (toutefois pas recommandé, les noms des paquets sont sensibles à la casse) ;
- [FIX] quelques actualisation des exemples de configuration `wapt-get.ini.tpl`
- [FIX] la compilation du **waptagent** échoue si les clés / certificats existent déjà mais que le certificat a été supprimé de `C:\wapt\ssl` ;
- [FIX] affichage dans la barre des tâches de la fenêtre de login (pour permettre en particulier l'autofill par des gestionnaires de mot de passe) ;

### 81.4.43 WAPT 1.3.9.3 (2017-04-11)

- [FIX] Argument `shell = True` was not explicitly passed to the underlying function as it occurred on previous versions.

### 81.4.44 WAPT 1.3.9 (2017-03-03)

#### Fixes

- [FIX] update code to follow more PEP8 recommandations ;
- [FIX] upgradedb locks sqlite database issue ;
- [FIX] Fix broken DNS SRV record discovery ;
- [FIX] Fix unicode handling of signer / CN / organization in certificates ;
- [FIX] Unzipped netifaces module ;

### wapt-get

- [NEW] Expands wildcards args for **install**, **show**, **build-package**, **sign-package**;
- [FIX] Fix **show-params** wapt-get command;
- [FIX] Fix **register** with description not working on some computers;
- [FIX] Fix broken `-c -config` option;

### Added setuphelpers functions

- [NEW] **reg\_key\_exists**;
- [NEW] **reg\_value\_exists**;
- [NEW] **run\_powershell**;
- [NEW] **remove\_metroapp**;
- [NEW] **local\_users\_profiles**;
- [NEW] **get\_profiles\_users**;
- [NEW] **get\_last\_logged\_on\_user**;
- [NEW] **get\_user\_from\_sid**;
- [NEW] **get\_profile\_path**;
- [NEW] **wua\_agent\_version**;
- [NEW] **local\_admins**;
- [NEW] **local\_group\_memberships**;
- [NEW] **local\_group\_members**;

### Modified helpers

- [IMP] command `:run` : explicit default values for **run** command help in **PyScripter**. Added `return_stderr` argument (overloaded str object);
- [FIX] **run\_notfatal** : fix unicode issue in use wmi module for **wmi\_info\_basic** instead of **wmic** shell command;
- [IMP] **make\_path** : improved when first argument is a drive. Be smart if an argument is a callable;
- [FIX] **CalledProcessError** : restored command `:CalledProcessError` alias;
- [ADD] **host\_infos** : added `profiles_users`, `last_logged_on_user`, `local_administrators`, `wua_agent_version` attributes;
- [IMP] **ensure\_unicode** : return None if None, for bytes strings, try utf8 decoding before system locale decoding;

### Console WAPT

- [FIX] restore allowed lowercase/uppercase package naming;
- [ADD] 4 host popup menu actions :
  - *Computer Mgmt*;
  - *Computer Users*;
  - *Computer Services*;
  - *RemoteAssist*;
- [FIX] fixed other issues in the WAPT console :
  - Don't search host while typing;
  - utf8 search (accents...);
  - utf8 compare;
  - try to get localized versions of special folders;

## Setup

- [ADD] **waptpythonw.exe** binary in distribution for console less python scripts (to avoid having **cmd.exe** windows popping up when invoking a python script);
- [FIX] change default wapt templates URL to <https://store.wapt.fr/wapt>;
- [FIX] when upgrading, (full **waptagent.exe** install) remove stalled **waptagent.exe** installs;

## 81.4.45 WAPT 1.3.8.2 (2016-11-18)

### Security

- [SEC] Fix inheritance of rights on wapt root folder for Windows 10 during setup when installed in C:\wapt. On Windows 10, **cacls.exe** does not work and does not remove « Authenticated Users » from C:\wapt. **cacls.exe** has been replaced by **icacls.exe** :
- on pre-wapt 1.3.7 systems, you can fix this by running the following command, or upgrade to wapt 1.3.8 (you may check **icacls.exe c:wapt /inheritance:r**)
- This can be achieved with a GPO, or a wapt package
- [IMP] in next versions of WAPT, the default install path of wapt will be changed from root folder C:\wapt to a more standard C:\Program Files (x86)\wapt.
- [IMP] By default, **waptsetup.exe** / **waptsetup-tis.exe** do not distribute certificates to avoid to deploy directly packages from Tranquil IT. **waptagent.exe** by default distributes the certificates that are installed on the mangement desktop creating the **waptagent**.

### Core changes

- [IMP] The database structure has changed between 1.3.8 and 1.3.8.2 to include additional attributes from packages : *signer*, *signer\_fingerprint*, *locale*, and *maturity*. *signer* and *signer\_fingerprint* are populated when signing the package to identify the origin. This means local WAPT database is upgraded when first starting WAPT 1.3.8.2 and this is not backward compatible;
- [IMP] Installers have a limited set of options, the most common use of WAPT is privileged;
- [ADD] 3 new parameters for the **waptexit** policy behavior : *hiberboot\_enabled*, *max\_gpo\_script\_wait*, *pre\_shutdown\_timeout*. These parameters are not set by default and should be added to **wapt-get.ini** [*global*] section if needed;
- [IMP] Use user's **waptconsole.ini** configuration file instead of **wapt-get.ini** for the commands targeted to package development (*sources*, *make-template*, *make-host-template*, *make-group-template*, *build-package*, *sign-package*, *build-upload*, *duplicate*, *edit*, *edit-host*, *upload-package*, *update-packages*). This avoids the need to write these parameters in **wapt-get.ini** on the development workstation. These parameters are not shared across multiple users on same machine. One use case is to allow multiple profiles (key, upload location) depending on the maturity of package (development, test, production...);

### Setuphelpers

- [ADD] helper functions **dir\_is\_empty**, **file\_is\_locked**, **service\_restart** and **WindowsVersions** class
- [IMP] Added referer and *user\_agent* in **wget** and **wgets**
- [IMP] run function : define stdin as PIPE to avoid lockup process waiting for input or error like unable to duplicate handle when using for example powershell
- [IMP] Version class : try to compare version using at least *Version.members\_count*
- [FIX] encoding fixes for registry functions, fix encoding for *registry\_setstring* key name
- [FIX] **install\_exe\_if\_needed** : don't check *uninstall\_key* or *min\_version* if not provided
- [FIX] **install\_exe\_if\_needed** and **install\_msi\_if\_needed** version check if *-force*
- [UPD] Check version and uninstall key after install with **install\_exe\_if\_needed** and **install\_msi\_if\_needed**

- [UPD] inventory includes informations from `WMI.Win32_OperatingSystem`
- [ADD] **get\_disk\_free\_space** helper function
- [UPD] check free disk space when downloading with **wget**. Check http status before.
- [UPD] Version class : `Version("7")<Version("7.1")` should return True

### wapt-get

- [ADD] 2 commands to get server SSL certificate and activate the certificate checking when using https with waptserver
- [FIX] **get\_sources** to allow svn checkout of a new package project
- [FIX] **register** problems with some BIOS with bitmaps
- [UPD] Check uninstall key after package install if `uninstallkey` is provided
- [FIX] added compatibility OS in `manifest` file for **wapt-get** and **waptconsole** version windows
- [FIX] erroneous error messages for **session-setup** in the WAPT console
- [UPD] add « `pattern` » parameter to `all_files` function
- [FIX] Install Date incorrectly registered by **register\_uninstall**
- [ADD] **user\_local\_appdata** function
- [ADD] add the `signer CN` and `signer_fingerprint` to `control` file when building package
- [ADD] add control attributes `min_wapt_version` to trigger an exception if `Package` requires a minimum level of libraries. The version is checked againts **setuphelpers.py** “s `__version__`” attribute.
- [ADD] `authorized_certificates` attribute is sent to the WAPT server. It contains the list of host’s signer certificates distributed on the host
- [FIX] When signing, check if WAPT zip file has already a signature file. (python zipfile can not replace the file inline)

### waptservice

- [ADD] Show *All Versions* checkbox in *Available Packages* page
- [UPD] Skin updated
- [ADD] *Filter* searchbox for available packages

### waptconsole

- [ADD] Add *NOT* checkbox for keywords search in **waptconsole** to search for hosts NOT having a specific package or software...
- [FIX] fix integer limit for grid display of package size, use int64 for size of packages in **waptconsole**.
- [UPD] don’t list packages of section « *restricted* » in local webservice available packages list
- [UPD] *Common Name* attribute should be populated now, so that signer identity is not None in package `control` file.
- [ADD] signer’s identity column in packages grid
- [FIX] escape quotes in package’s description
- [ADD] Check **waptagent.exe** version against **waptsetup-tis** version at **waptconsole** startup.
- [UPD] try to display a *progress* dialog at **waptconsole** startup
- [FIX] company not set when building customized **waptagent.exe**
- [ADD] initialize Organization in **waptagent.exe** build with CN from certificate.

**waptexit**

- [UPD] some text introduction changes

**waptray**

- [NEW] Limit trayicon balloon popup when Windows version is above Windows 7 or if `notify_user = 0` in `wapt-get.ini`

**waptserver**

- [UPD] Use broadcast address on interface for wakeonlan call
- [FIX] remove the check of wapt server password which prevents the proper registration of **waptserver** on Windows.
- [UPD] when upgrading, reuse existing `waptserver.ini` file if it already exists, don't overwrite `server_uuid` and ask for password reset if it already exists

**waptdeploy waptupgrade**

- [FIX] **waptdeploy** not working on WinXP removed `DisableWow64FileSystemRedir` on **runtask**.
- [FIX] **waptupgrade** : Missing quotes for system account on Windows XP

**Libraries**

- [ADD] BeautifulSoup for wapt packages auto updates tasks
- [UPD] **winsys** library update to "1.0b1"

**81.4.46 WAPT 1.2.3.2 (2015-05-05)**

- [ADD] `UUID` parameter for direct requests to hosts from the WAPT Server;
- [ADD] allow host to refuse request if not right target (if ip has changed since last **update\_status** for example)
- [ADD] fallback on waptserver `usage_statistics` if `mongodb` lacks aggregate support
- [IMP] register host on server in `postconf` using **waptservice** http instead of command line **wapt-get**

**81.4.47 WAPT 1.2.2 (2015-04-22)**

- [ADD] **reset-uuid** and **generate-uuid** for <https://roundup.tranquil.it/wapt/issue421> duplicated `UUID` issues
- [IMP] mass hosts delete, added delete hosts package action. server `>=1.2.2` only : <https://roundup.tranquil.it/wapt/issue433>
- [ADD] read the docs theme for sphinx setuphelpers API documentation. WIP <https://roundup.tranquil.it/wapt/issue427>
- [IMP] doc updates
- [ADD] `api/v1/hosts_delete` method
- [ADD] **need\_install**, **install\_exe\_if\_needed**, **install\_msi\_if\_needed** functions to setuphelpers
- [ADD] parameters for **waptdeploy**.

## 81.4.48 WAPT 1.2.1 (2015-03-26)

### Console WAPT

- [ADD] combobox for filtering on groups in **waptconsole**.
- [ADD] *Add ADS Groups as packages* action to WAPT host selection popup menu
- [ADD] **cleancache** action to clean local waptconsole packages cache
- [ADD] added **notify\_server** on network reconfiguration if **waptserver** is available;
- [IMP] column *groups* shows only host's direct dependencies with package's section == « group » instead of all direct dependencies.
- [ADD] optional anonymous statistics (nb of machines, nb of packages, age of updates...) sent to Tranquil IT to document the communication around WAPT (sent by **waptconsole** at most every 24h)
- [IMP] improved mass hosts delete,
- [ADD] delete hosts package action. server >=1.2.2 only : <https://roundup.tranquil.it/wapt/issue433>
- [IMP] big packages uploads (write uploaded packages by chunk) (but still some issues on 32bits servers due to **uwsgi**)
- [IMP] display version of mismatch when editing package
- [FIX] host's packages not saved when some dependencies don't exist anymore
- [FIX] restore working *Cancel running task* button
- [FIX] canceling subprocesses not working in freepascal apps (when waiting for **InnoSetup** compile for example)

### wapt-get / waptservice

- [ADD] **reset-uuid** and **generate-uuid** for <https://roundup.tranquil.it/wapt/issue421> duplicated *UUID* issues
- [IMP] **find\_wapt\_repo\_url** processus to avoid waiting for all repos if one repo is ok (improved response time in buggy networks)
- [IMP] windows DNS resolver in wapt client (python part) instead of pure python resolver. Should reduce issues when multiple network cards or inactive network connections.
- [IMP] changed priority of server discovery using SRV dns records. -> first priority ascending and weight descending. -> comply with standards.
- [FIX] solved some issues with **SQLite** and threads in local **waptservice**
- [IMP] explicit transaction handling and *isolation\_level = None* for local waptDB (to try to avoid locks)
- [IMP] teardown handler for **waptservice** to commit or rollback thread local connections
- [FIX] for waptrepo detection in freepascal parts : same processus as python part.
- [FIX] for **edit\_package** when supplying a wapt filename instead of package request

### Setuphelpers

- [ADD] read the docs theme for sphinx setuphelpers API documentation. WIP <https://roundup.tranquil.it/wapt/issue427>
- [ADD] **\_all\_** list to avoid importing unnecessary names in **setup.py** modules. Now only functions defined in **setuphelpers** are available when importing **setuphelpers**. This can break some WAPT packages if names were indirectly imported through **setuphelpers** module.
- [ADD] **need\_install**, **install\_exe\_if\_needed**, **install\_msi\_if\_needed** functions to **setuphelpers**
- [ADD] **local\_desktops** function
- [FIX] version class instances accept to be compared to str
- [REM] **processnames\_list** which is unused in **setuphelpers**
- [ADD] **add\_ads\_groups** and **get\_computer\_groups** to **waptdevutils.py**
- [FIX] **run** helper
- [FIX] **on\_write** callback not working
- [FIX] **TimeoutExpired** not formatted properly
- [FIX] use closure for registry keys

## Waptdeploy

- [IMP] **waptdeploy** with more command line options (in particular tasks to merge to default innosetup selected tasks)
- [FIX] waptrepo detection using dns records

## Install

- [FIX] **waptagent** upload error on windows
- [FIX] debian packages should work for Jessie
- [IMP] **copytree2** for **waptupgrade**
- [FIX] trap exception for version check on copy of **exe** and **dll**
- [FIX] **mongodb-server** version should be  $\geq 2.4$

## 81.4.49 WAPT-1.1.1 (2015-02-26)

### Console WAPT

- [IMP] the loading of the main grid has been optimized ; only configured columns are displayed ;
- [IMP] the WAPT server detects the hosts whose **waptservice** is listening. Their *Reachable* status is shown with a green / grey indicator ;
- [IMP] the WAPT package to upgrade WAPT on hosts (???-waptupgrade.wapt) is generated by the WAPT console at the same time as the WAPT agent installer (**waptagent.exe**), the two files are then uploaded on the WAPT server ;
- [ADD] the package dependencies of each host are displayed in the grid. This allows to see what hosts have no package ;
- [ADD] possibility to trigger available package upgrades on hosts that are listening from the WAPT console. In that case, the host sends its status to the WAPT server after the upgrade ;
- [ADD] possibility to filter hosts in the WAPT console according to their upgrade status or whether they are « reachable » or not,
- [ADD] when packages are flagged for install but are not yet installed on a host, they appear with a blue « + » indicator. It is then possible to force the immediate install of the package with a right-click ;

### Waptservice

- [ADD] cleaning of the cache on the hosts after each successful upgrade ;

### Waptserver

- [ADD] the versions of the WAPT agent, WAPT Server are shown in the main web page of the WAPT Server (with a red indicator if there is a problem) ;

### Packaging

- [ADD] functions to **setuptools** to manage shortcuts :
- **remove\_desktop\_shortcut** ;
- **remove\_user\_desktop\_shortcut** ;
- **remove\_programs\_menu\_shortcut** ;
- **remove\_user\_programs\_menu\_shortcut** ;

### Installation

- [IMP] verification of used ports during the post-configuration of WAPT Server on a Windows machine ;

### Webservices

- [IMP] the **waptserver** no longer listen on 8080 port by default.

The Apache frontal web server listens in HTTP and HTTPS and relays action calls to the python **waptservice** that only listens locally.

It is therefore necessary to update `wapt-get.ini` files on WAPT agents and to replace `wapt_server = http://srvwapt.mydomain.lan:8080` with `wapt_server = https://srvwapt.mydomain.lan`

If you can not make that change to your WAPT agents, it is possible to return to the previous behavior.

On Debian, edit the file `/opt/wapt/waptserver/waptserver.ini`, and in the `[uwsgi]` section, put :

```
http-socket = 0.0.0.0 :8080
```

On Windows, edit `C:\waptwaptserver\waptserver.ini` and replace :

```
server = Rocket(("127.0.0.1", port), "wsgi", {« wsgi_app » :app})
```

with :

```
server = Rocket(("0.0.0.0", port), "wsgi", {« wsgi_app » :app})
```

The repository may stay in HTTP on port 80.

The calls to the WAPT Server are authenticated, but it is advised to restrict access to authorized sub-networks with a firewall.

- [IMP] json calls to the webservice of the WAPT Server are now standardized ;
- [IMP] when launching **update / upgrade / remove / forget / tasks\_status** actions from the WAPT console, the IP address of the host is no longer sent, but instead its *UUID*, and it is the WAPT Server that finds the IP address and the port to use ; et c'est le serveur wapt qui s'occupe de déterminer quelle IP / port utiliser ;
- [ADD] verification in the WAPT console that the version of the WAPT Server is sufficient ;
- [ADD] the timeout to connect to WAPT agents and read the data are configurable in `waptserver.ini` ;



#### 81.4.50 WAPT-1.0 (2015-01-31)

— [ADD] first public version of WAPT



---

## Accord de licence d'utilisateur final WAPT

---

This Enterprise End User License Agreement (this “Agreement”) is made between Tranquil IT (“Company”), and you (“Customer,” “you” or “your”), for the use of WAPT Enterprise and WAPT packages (the “Software”), as described. By downloading, installing or using the Software accompanying this Agreement, you acknowledge that you have reviewed and accept this Agreement. If you are agreeing to this Agreement as an individual, “Customer”, “you” and “your” refers to you individually. If you are agreeing to this Agreement as a representative of an entity, you represent that you have the authority to bind that entity, and “Customer”, “you” and “your” refers to that entity specified in the sales confirmation or other agreement to purchase the enterprise license to the Software. If you do not agree with all of the terms of this Agreement, do not download or otherwise use the Software.

### 82.1 Définitions

Dans le présent accord, les termes en majuscules ont la signification suivante :

- « AFFILIÉ » signifie toute entité qui, directement ou indirectement, par un ou plusieurs intermédiaires, contrôle, est contrôlée par, ou est sous contrôle commun avec l’entité concernée. Aux fins de la présente définition, on entend par « contrôle » la possession directe ou indirecte du pouvoir de diriger ou de faire diriger la gestion et les politiques d’une entité, que ce soit par la propriété de titres avec droit de vote, par contrat ou autrement.
- « OBJECTIFS AUTORISÉS » désigne les objectifs techniques ou commerciaux internes du client.
- « SYSTÈME CLIENT » signifie le(s) site(s) web interne(s) du Client, les serveurs et autres équipements et logiciels, y compris, sans limitation, les dispositifs et systèmes mobiles basés sur des émulations virtuelles ou logiques.
- « DATE DE LIVRAISON » désigne la date, indiquée dans le texte applicable, à laquelle il est prévu que le logiciel soit mis à la disposition du client.
- « DOCUMENTATION » désigne les instructions d’utilisation et les fichiers d’aide imprimés, sur papier, électroniques ou en ligne, généralement mis à disposition par la Société pour l’utilisation du Logiciel, tels qu’ils peuvent être mis à jour de temps en temps par la Société.
- « DROITS DE PROPRIÉTÉ INTELLECTUELLE » : tous les droits de propriété intellectuelle ou droits de propriété similaires, y compris (a) les droits de brevet et les modèles d’utilité, (b) les droits d’auteur et les droits relatifs aux bases de données, (c) les marques, les noms commerciaux, les noms de domaine et l’habillage commercial, ainsi que l’achalandage qui y est associé, (d) les secrets commerciaux, (e) les travaux sur les masques et (f) les droits relatifs aux dessins et modèles industriels ; dans

chaque cas, y compris tout enregistrement, toute demande d'enregistrement, tout renouvellement et toute prolongation de ces droits dans toute juridiction du monde.

- “OPEN SOURCE SOFTWARE” means all software that is available under the GNU Affero General Public License (AGPL), GNU General Public License (GPL), GNU Lesser General Public License (LGPL), Mozilla Public License (MPL), Apache License, BSD licenses, or any other license that approved by the Open Source Initiative ([www.opensource.org](http://www.opensource.org)).
- « CONFIRMATION DE VENTE » signifie les documents de commande pour les services et les licences pour les logiciels achetés à la Société qui sont conclus ou autrement mutuellement acceptés par écrit en vertu des présentes par les parties de temps à autre, y compris les modifications, les suppléments et les addenda à ceux-ci. En cas d'incohérence ou de conflit entre une confirmation de vente et le présent contrat, le présent contrat régit, à moins que la confirmation de vente n'identifie spécifiquement par référence à une Section, la disposition que cette confirmation de vente modifie, et alors cette modification s'appliquera uniquement à cette confirmation de vente. Les conditions générales énoncées dans une confirmation de vente, une confirmation de vente ou une facture (à l'exception des détails du logiciel fourni, du prix, de la quantité de licences, de la durée de l'abonnement, de la date de livraison et d'autres détails de livraison qui ne sont pas préimprimés et qui sont incompatibles avec les conditions du présent contrat) ne sont pas valables et n'ont aucun effet, et aucune pratique commerciale, usage commercial ou mode d'exécution n'est pertinent pour expliquer ou modifier une condition exprimée dans le présent contrat.
- « LOGICIEL » désigne les logiciels WAPT Enterprise et WAPT spécifiés dans une confirmation de vente et toutes les mises à jour de la Société que la Société fournit au Client conformément aux services de support que le Client est en droit de recevoir en vertu du présent accord, sous forme de code objet ou sous forme de texte. Pour tous les besoins du présent accord, le terme « logiciel » exclut tout logiciel open source et toutes les offres de tiers, telles que les logiciels de tiers, le contenu et autres actifs virtuels et numériques.
- « SERVICES DE SUPPORT » désigne les services de support et de maintenance fournis par la Société en vertu de la Section *Services de maintenance et de support* ci-dessous.
- « OFFRES DE TIERS » désigne certains logiciels ou services fournis ou exécutés par des tiers qui sont nécessaires au fonctionnement du Logiciel.
- « MISES À JOUR » désigne les corrections de bogues, les correctifs et les versions de maintenance du logiciel dans la mesure où ils sont généralement mis à la disposition de ses licenciés par la Société.
- « DISPOSITIFS » désigne tout dispositif du Client ou de ses affiliés sur lequel l'agent WAPT Enterprise est installé.

## 82.2 Commandes, octroi de licence, clé d'activation et garantie de performance

### 82.2.1 Commandes

Sous réserve des termes et conditions contenus dans le présent contrat, vous pouvez acheter des licences pour des appareils afin d'utiliser le logiciel conformément à une confirmation de vente. Les utilisateurs ne peuvent pas transférer les licences et ne peuvent pas installer ou utiliser le logiciel sur un nombre de dispositifs uniques supérieur à celui indiqué dans la confirmation de vente. Le Client accepte que ses achats en vertu des présentes ne soient pas conditionnés par la livraison de toute fonctionnalité ou fonction future, ni dépendants de tout commentaire public oral ou écrit fait par la Société concernant toute fonctionnalité ou fonction future.

## 82.2.2 Octroi de licences et limitations de licences

Sous réserve des termes et conditions du présent contrat, la société vous accorde par les présentes une licence non exclusive, non transférable, non cessible, révocable et limitée pour l'utilisation du logiciel et de toute documentation, sans droit de sous-licence du logiciel, uniquement pour les besoins autorisés du client et non au profit d'une autre personne ou entité. Vous ne pouvez pas, ni permettre à un tiers de, prêter, louer, distribuer, transférer ou rendre disponible le logiciel à un tiers. Sauf si cela est nécessaire pour votre utilisation personnelle du Logiciel, vous ne pouvez pas copier le Logiciel, en tout ou en partie, à l'exception des composants tiers open source énumérés dans la Section *Composants logiciels tiers* dont les licences donnent le droit de copier ces composants. Votre utilisation du Logiciel peut être soumise à certaines limitations, comme indiqué dans la Confirmation de vente. Ces limitations seront spécifiées soit dans la confirmation de vente, soit dans la documentation. À l'exception de ce qui est expressément accordé dans le présent contrat, aucune autre licence n'est accordée au client, que ce soit de manière expresse, implicite ou par préclusion. Tous les droits non accordés dans le présent accord sont réservés par la société.

## 82.2.3 Restrictions

Vous ne pouvez pas, directement ou indirectement, et vous ne permettrez à aucun utilisateur ou tiers de :

- Faire de l'ingénierie inverse, décompiler, désassembler ou tenter de toute autre manière de découvrir le code source ou les idées ou algorithmes sous-jacents du logiciel.
- Modifier, traduire ou créer des travaux dérivés basés sur tout élément du logiciel ou de toute documentation connexe.
- Louer, donner en location, distribuer, vendre, revendre, céder ou transférer de toute autre manière vos droits d'utilisation du logiciel.
- Utiliser le logiciel à des fins de partage de temps ou autrement au profit de toute personne ou entité autre que le Client.
- Supprimer toute mention de propriété de la documentation.
- Publier ou divulguer à des tiers toute évaluation ou analyse comparative du logiciel sans le consentement écrit préalable de la Société.
- Utiliser le logiciel dans un but autre que celui auquel il est destiné.
- Interférer avec ou perturber l'intégrité ou les performances du logiciel ; ou
- Tenter d'obtenir un accès non autorisé au logiciel ou aux systèmes ou réseaux de l'entreprise.

## 82.2.4 Clé d'activation

Vous pouvez activer le Logiciel au moyen d'une clé d'activation fournie par le Client ou par la Société, comme indiqué dans la Confirmation de vente applicable.

## 82.2.5 Garantie de performances

La Société garantit que le logiciel fonctionnera dans les conditions matérielles décrites dans la documentation pendant une période de 30 jours à compter de la date d'achat de la commande (la « période de garantie »). Si le Client se rend compte que le logiciel ne fonctionne pas en conformité substantielle avec la documentation (un « défaut »), le Client doit fournir à la Société une notification écrite qui comprend une explication raisonnablement détaillée du défaut pendant la période de garantie. Si la Société est en mesure de reproduire le défaut dans son propre environnement d'exploitation, la Société fera des efforts commercialement raisonnables pour corriger rapidement le défaut ou fournir au Client un logiciel de remplacement avec des fonctionnalités substantiellement similaires.

**CE QUI PRÉCÈDE ÉNONCE LA RESPONSABILITÉ UNIQUE ET EXCLUSIVE DE LA SOCIÉTÉ ET LE RECOURS UNIQUE ET EXCLUSIF DU CLIENT POUR TOUT LOGICIEL DÉFECTUEUX.**

## 82.3 Confidentialité

### 82.3.1 Informations confidentielles

« informations confidentielles » : toutes les informations techniques et non techniques non publiques divulguées par une partie (la « partie émettrice ») à l'autre partie (la « partie destinataire ») sous quelque forme ou support que ce soit, oral, écrit, graphique ou électronique, en vertu du présent accord, qui sont marquées comme confidentielles et exclusives, ou que la partie émettrice identifie comme confidentielles et exclusives, ou qui, en raison de la nature des circonstances entourant la divulgation ou la réception, doivent être traitées comme des informations confidentielles et exclusives, notamment :

- Techniques, esquisses, dessins, modèles, inventions (brevetées ou non), savoir-faire, procédés, appareils, formules, équipements, algorithmes, programmes logiciels, documents sources de logiciels, API et autres œuvres de création (protégées ou non par le droit d'auteur).
- Informations concernant la recherche, les travaux expérimentaux, le développement, les détails et spécifications de la conception, l'ingénierie, les informations financières, les besoins d'approvisionnement, les achats, la fabrication, les listes de Clients, les prévisions commerciales, les plans et informations de vente et de merchandising.
- Les informations exclusives ou confidentielles de tout tiers qui pourrait divulguer ces informations à la partie émettrice ou à la partie réceptrice dans le cadre des activités de la partie émettrice ; et
- Les conditions du présent accord et toute confirmation de vente ou tout cahier des charges. Les informations confidentielles de la Société comprennent le logiciel, la documentation, les prix et les informations concernant les caractéristiques, les fonctionnalités ou les performances des licences bêta et des solutions non-GA. Les Informations Confidentielles comprennent également tous les résumés et abstracts des Informations Confidentielles.

### 82.3.2 Non-divulgation

Chaque partie reconnaît que dans le cadre de l'exécution du présent accord, elle peut obtenir les informations confidentielles de l'autre partie. La partie réceptrice doit, à tout moment, tant pendant la durée du présent accord qu'après celle-ci, faire des efforts raisonnables pour garder confidentielles et fiables toutes les informations confidentielles de la partie émettrice qu'elle reçoit. La partie réceptrice ne doit pas utiliser les informations confidentielles de la partie émettrice autrement que pour remplir les obligations de la partie réceptrice ou pour exercer les droits de la partie réceptrice en vertu du présent accord. Chaque partie s'engage à sécuriser et à protéger les Informations Confidentielles de l'autre partie avec le même degré de soin et d'une manière compatible avec la maintenance de ses propres Informations Confidentielles (mais en aucun cas moins que le soin raisonnable), et à prendre les mesures appropriées par instruction ou accord avec ses employés, affiliés ou autres agents qui sont autorisés à accéder aux Informations Confidentielles de l'autre partie pour satisfaire ses obligations en vertu de la présente Section. La partie réceptrice ne doit pas divulguer les informations confidentielles de la partie émettrice à toute personne ou entité autre que ses dirigeants, employés, affiliés et agents qui ont besoin d'accéder à ces informations confidentielles afin de réaliser l'intention du présent accord et qui sont soumis à des obligations de confidentialité au moins aussi strictes que celles énoncées dans le présent accord.

### 82.3.3 Exceptions aux informations confidentielles

Les obligations énoncées à la section *Non-divulgation* ne s'appliquent pas dans la mesure où les informations confidentielles comprennent des informations qui :

- Etait connu par la partie destinataire avant d'être reçu de la partie émettrice, soit par elle-même, soit par réception directe ou indirecte d'une source autre que celle ayant une obligation de confidentialité envers la partie émettrice.
- A été développé par la partie réceptrice sans utilisation des informations confidentielles de la partie émettrice ; ou
- Devient publiquement connu ou cesse d'être secret ou confidentiel, sauf en cas de violation du présent accord ou de toute obligation de confidentialité par la partie destinataire. Rien dans le présent accord n'empêche la partie destinataire de divulguer des informations confidentielles dans la mesure où la partie destinataire est légalement obligée de le faire par un organisme d'enquête gouvernemental ou judiciaire dans le cadre d'une procédure relevant de la compétence de cet organisme ; à condition toutefois qu'avant toute divulgation, la partie destinataire le fasse :

- Faire valoir le caractère confidentiel des informations confidentielles auprès de l'agence.
- Notifier immédiatement par écrit à la partie divulgatrice l'ordre ou la demande de divulgation de l'agence ; et (z) coopérer pleinement avec la partie divulgatrice pour se protéger contre une telle divulgation et obtenir une ordonnance de protection limitant la portée de la divulgation forcée et protégeant sa confidentialité.

### 82.3.4 Mesures d'injonction

Les parties conviennent que toute divulgation non autorisée d'informations confidentielles peut causer un préjudice immédiat et irréparable à la partie divulgatrice et que, dans le cas d'une telle violation, la partie destinataire aura le droit, en plus de tout autre recours disponible, de demander une injonction immédiate et d'autres mesures de redressement équitables, sans caution et sans qu'il soit nécessaire de démontrer des dommages pécuniaires réels.

## 82.4 Droits de propriété

### 82.4.1 Logicielle

Entre la Société et le Client, tous les droits, titres et intérêts relatifs au Logiciel et à tous les matériels, logiciels, éléments virtuels et autres contenus fournis ou mis à disposition en vertu des présentes, ainsi que toutes les modifications et améliorations qui y sont apportées, et toutes les suggestions, idées et commentaires proposés par le Client concernant ces éléments, y compris tous les droits d'auteur, droits de brevet et autres droits de propriété intellectuelle dans chacun des cas précités, appartiennent et sont conservés uniquement par la Société ou les concédants et fournisseurs de la Société, selon le cas.

### 82.4.2 Développements menés par Tranquil IT

Toutes les inventions, les œuvres de l'auteur et les développements conçus, créés, écrits ou générés par ou au nom de la Société, que ce soit seul ou conjointement, et tous les droits de propriété intellectuelle y afférents, sont la propriété unique et exclusive de la Société. Le Client convient que, dans la mesure où la propriété de toute contribution du Client ou de ses employés à la création du logiciel n'est pas, par effet de la loi ou autrement, dévolue à la Société, le Client cède et convient par les présentes de céder à la Société tous les droits, titres et intérêts relatifs à ces contributions, y compris, sans s'y limiter, tous les droits de propriété intellectuelle y afférents, sans qu'il soit nécessaire de procéder à une quelconque autre considération.

### 82.4.3 Autres garanties

Dans la mesure où les droits, titres et intérêts relatifs aux droits de propriété intellectuelle ne peuvent être cédés par le Client à la Société, le Client accorde par les présentes à la Société une licence exclusive, libre de redevances, transférable, irrévocable, mondiale et entièrement libérée (avec des droits de sous-licence par le biais de plusieurs niveaux de sous-licenciés) pour utiliser, pratiquer et exploiter pleinement ces droits, titres et intérêts non cessibles. Si la cession et la licence susmentionnées ne sont pas exécutoires, le Client accepte de renoncer à ces droits, titres et intérêts incessibles et non susceptibles de faire l'objet d'une licence et de ne jamais les faire valoir à l'encontre de la Société. Le Client accepte d'exécuter tout document ou de prendre toute mesure qui pourrait être raisonnablement nécessaire, ou que la Société pourrait raisonnablement demander, pour parfaire la propriété de ces droits. Si le Client ne peut pas ou ne veut pas exécuter un tel document ou prendre une telle mesure, la Société peut exécuter ce document et prendre cette mesure au nom du Client en tant qu'agent et mandataire du Client. La nomination ci-dessus est considérée comme un pouvoir couplé à un intérêt et est irrévocable.

## 82.5 Obligations du Client

Le Client est responsable de :

- Obtenir, déployer et maintenir le système informatique du Client et tout le matériel informatique, les logiciels, les périphériques et l'équipement de réseau nécessaires pour que le Client, ses affiliés et leurs utilisateurs respectifs puissent utiliser le Logiciel ; et
- Payer tous les frais de tiers et des frais d'accès encourus en rapport avec ce qui précède. Sauf indication contraire dans le présent contrat ou dans une confirmation de vente, la Société ne sera pas responsable de la fourniture de matériel ou d'autres équipements au Client dans le cadre du présent contrat.

## 82.6 Services de maintenance et de support

### 82.6.1 Maintenance et support

Sous réserve des termes et conditions du présent accord (y compris le paiement des frais applicables, le cas échéant), la Société fera des efforts commercialement raisonnables pour fournir les « services de support » suivants :

- Une assistance technique par forum et liste de diffusion (dans un délai raisonnable) si le service Enterprise Support n'a pas été souscrit.
- Une assistance technique par téléphone si le service Enterprise Support a été souscrit et.
- Bug fixes. Support Services may include Updates generally issued by Company to customers, but do not include new versions of the Software, for which Company may charge a fee. In no event will Support Services apply with respect to Third Party Offerings. Company does not guarantee that it will provide Support Services for versions other than for then-current versions of the Software.

### 82.6.2 Durée du support ; résiliation

Sauf indication contraire dans un bon de commande, la Société fournira des services d'assistance à partir de la date à laquelle le logiciel est téléchargé pour la première fois en vue de son utilisation et aussi longtemps que le Client maintient une version du logiciel à jour (la « période d'assistance »).

## 82.7 Audit

La Société a le droit d'examiner l'utilisation du logiciel par le Client et d'entrer dans les installations et les locaux du Client uniquement pour vérifier que le nombre d'ordinateurs avec le logiciel utilisé par le Client n'excède pas le nombre de licences accordées au Client en vertu du présent contrat. Toute visite aux installations du Client en vertu de la présente Section sera soumise aux règlements sur place du Client et aura lieu à un jour et une heure convenus mutuellement, au plus tôt 10 jours après notification de la Société. Alternativement, la Société peut demander au Client de fournir un rapport écrit sur l'identité et l'emplacement des ordinateurs avec le logiciel (détaillé sur une base mensuelle) afin de vérifier la conformité avec la licence accordée dans le présent contrat. Si un audit révèle une utilisation excessive des licences, la Société émettra une facture pour le nombre de licences égal au nombre d'ordinateurs excédentaires au tarif en vigueur pour le logiciel et le Client devra payer cette facture dans les trente (30) jours suivant la date de facturation. La Société paiera les coûts de l'audit à moins que cet audit ne révèle que des ordinateurs supplémentaires sont exploités par le Client sans la permission de la Société, auquel cas les coûts de l'audit seront payés par le Client.



## 82.8 Déclarations et garanties ; Clause de non-responsabilité

### 82.8.1 Représentations et garanties mutuelles

Chaque partie déclare, garantit et s'engage à ce que :

- Elle a les pleins pouvoirs et l'autorité de conclure le présent accord et de s'acquitter des obligations qui en découlent, sans qu'il soit nécessaire d'obtenir des consentements, des approbations ou des immunités qui ne le sont pas encore ; et
- Son acceptation et son exécution au titre du présent accord ne doivent pas enfreindre un accord oral ou écrit avec un tiers ni une obligation due par lui à un tiers de garder des informations ou des documents confidentiels ou en confiance.

### 82.8.2 Exclusion de garantie

À L'EXCEPTION DE LA GARANTIE LIMITÉE PRÉVUE CI-DESSUS AU POINT 2.5, LE LOGICIEL EST FOURNI « TEL QUEL », SANS GARANTIE D'AUCUNE SORTE, EXPRESSE OU IMPLICITE, Y COMPRIS, MAIS SANS S'Y LIMITER, TOUTE GARANTIE DE NON-CONTREFAÇON, DE PERFORMANCE, DE QUALITÉ MARCHANDE OU D'ADÉQUATION À UN USAGE PARTICULIER. VOUS ASSUMEZ L'INTÉGRALITÉ DES RISQUES LIÉS AU LOGICIEL EN CE QUI CONCERNE LA QUALITÉ, LES PERFORMANCES, LA PRÉCISION ET LES EFFORTS SATISFAISANTS. AUCUNE INFORMATION OU CONSEIL ORAL OU ÉCRIT DONNÉ PAR L'ENTREPRISE NE CRÉE DE GARANTIE. LA SOCIÉTÉ NE GARANTIT PAS QUE LE LOGICIEL RÉPONDRA À VOS EXIGENCES, FONCTIONNERA SANS INTERRUPTION OU SERA EXEMPT D'ERREURS. CERTAINES JURIDICTIONS N'AUTORISENT PAS L'EXCLUSION DES GARANTIES IMPLICITES OU DES LIMITATIONS TELLES QUE SPÉCIFIÉES ICI ET DANS LA MOINDRE MESURE AUTORISÉE PAR LA LOI, CES EXCLUSIONS ET LIMITATIONS PEUVENT NE PAS VOUS CONCERNER. AUCUN AGENT DE LA SOCIÉTÉ N'EST AUTORISÉ À MODIFIER OU À ÉTENDRE LES GARANTIES DE LA SOCIÉTÉ TELLES QU'ELLES SONT ÉNONCÉES DANS LE PRÉSENT DOCUMENT. LA SOCIÉTÉ NE GARANTIT PAS QUE :

- THE USE OF THE SOFTWARE WILL BE SECURE, TIMELY, UNINTERRUPTED OR ERROR-FREE OR OPERATE IN COMBINATION WITH ANY NON-SUPPORTED HARDWARE, SOFTWARE, SYSTEM OR DATA.
- THE SOFTWARE WILL MEET CUSTOMER'S REQUIREMENTS OR EXPECTATIONS ; OR
- THE SOFTWARE WILL BE ERROR-FREE OR THAT ERRORS OR DEFECTS IN THE SOFTWARE WILL BE CORRECTED. THE SOFTWARE MAY BE SUBJECT TO LIMITATIONS, DELAYS, AND OTHER PROBLEMS INHERENT IN THE USE OF THE INTERNET AND ELECTRONIC COMMUNICATIONS. COMPANY IS NOT RESPONSIBLE FOR ANY DELAYS, DELIVERY FAILURES, OR OTHER DAMAGES RESULTING FROM SUCH PROBLEMS.

## 82.9 Indemnisation ; limitation de la responsabilité

### 82.9.1 Indemnisation de la Société

#### Généralités

Pendant la période où le Client maintient une version actuelle du Logiciel, la Société défendra le Client et ses affiliés (les « Parties indemnisées par le Client ») contre toutes les actions, procédures, réclamations et demandes d'un tiers (une « Réclamation de tiers ») alléguant que le Logiciel viole un droit d'auteur ou détourne un secret commercial. Les obligations de la Société en vertu de la présente section sont conditionnées par (i) la notification rapide par écrit à la Société de toute réclamation en vertu de la présente section, (ii) le droit exclusif de la Société de contrôler la défense et le règlement de la réclamation, et (iii) la fourniture par le Client de toute l'assistance raisonnable dans la défense de cette réclamation. En aucun cas, le client ne doit régler une réclamation sans l'accord écrit préalable de la société. Le client peut, à ses propres frais, engager un avocat séparé pour le conseiller sur une réclamation, sous réserve du droit de la société de contrôler la défense et le règlement ;

### Mitigation

Si une réclamation que la société est obligée de défendre a eu lieu, ou selon la détermination de la société, est susceptible d'avoir lieu, la société peut, à sa seule discrétion et à son gré et à ses frais :

- Obtenir pour le Client le droit d'utiliser le logiciel.
- Substituer à ce logiciel une fonctionnalité équivalente et non contrefaisante.
- Modifier le logiciel pour le rendre non contrefaisant et fonctionnellement équivalent ; ou
- Mettre fin au présent accord.

### Exclusions

Nonobstant toute disposition contraire du présent accord, les obligations qui précèdent ne s'appliquent pas à une action en contrefaçon si celle-ci découle d'une plainte :

- L'utilisation du logiciel en combinaison avec tout logiciel, matériel, réseau ou système non fourni par la société lorsque la prétendue infraction se rapporte à cette combinaison.
- Toute modification ou altération du logiciel autre que par la société.
- La poursuite de l'utilisation du logiciel par le client après que la société lui ait notifié l'arrêt de l'utilisation en raison d'une plainte pour infraction.
- L'utilisation de logiciels à source ouverte.
- La violation par le client du droit applicable ; et
- Système du client.

### Seul remède

CE QUI PRÉCÈDE INDIQUE L'ENTIÈRE RESPONSABILITÉ DE LA SOCIÉTÉ EN CE QUI CONCERNE LA VIOLATION DE TOUTE PROPRIÉTÉ INTELLECTUELLE OU DE TOUT DROIT DE PROPRIÉTÉ PAR LE LOGICIEL OU AUTRE, ET LE CLIENT RENONCE EXPRESSÉMENT PAR LES PRÉSENTES À TOUTE AUTRE RESPONSABILITÉ OU OBLIGATION DE LA SOCIÉTÉ À CET ÉGARD.

## 82.9.2 Indemnisation des Clients

Le Client doit défendre la Société et ses affiliés, les concédants de licence et leurs dirigeants, administrateurs et employés respectifs (« parties indemnisées par la Société ») contre toute réclamation de tiers qui découle de ou se rapporte à :

- L'utilisation ou la prétendue utilisation du logiciel par le Client autre que celle autorisée par le présent contrat.
- Le Client ou les utilisateurs de ses affiliés utilisent le logiciel en violation de toute loi ou réglementation applicable, ou des droits de propriété intellectuelle ou autres droits d'un tiers ; ou
- Découlant de la survenance de l'une des exclusions énoncées dans la section *Exclusions*. Le client doit payer tous les dommages, frais et dépenses, y compris les honoraires et frais d'avocats (que ce soit par un règlement ou une décision judiciaire finale) payés au tiers qui introduit une telle réclamation. Les obligations du client en vertu de la présente section *Indemnisation des Clients* sont conditionnées :
  - Le client est rapidement informé par écrit de toute réclamation au titre de la présente section *Indemnisation des Clients*.
  - Le client a le droit unique et exclusif de contrôler la défense et le règlement de la réclamation ; et
  - La société fournit toute l'assistance raisonnable (aux frais du client et sur demande raisonnable) pour la défense de cette réclamation.

En aucun cas, la société ne peut régler une réclamation sans l'approbation écrite préalable du client, laquelle ne doit pas être refusée, retardée ou conditionnée de manière déraisonnable. La société peut, à ses propres frais, engager un avocat séparé pour conseiller la société concernant une réclamation de tiers et pour participer à la défense de la réclamation, sous réserve du droit du client de contrôler la défense et le règlement.

### 82.9.3 Limitation de la responsabilité et des dommages ; exclusion des recours et des dommages

EN AUCUN CAS, LA SOCIÉTÉ NE SERA RESPONSABLE DES DOMMAGES ACCESSOIRES, CONSÉQUENTIELS, PUNITIFS OU SIMILAIRES DE QUELQUE NATURE QUE CE SOIT, DÉCOULANT DU LOGICIEL OU DU PRÉSENT CONTRAT. EN AUCUN CAS, LA RESPONSABILITÉ GLOBALE DE LA SOCIÉTÉ DÉCOULANT DU LOGICIEL OU DU PRÉSENT CONTRAT OU Y ÉTANT LIÉE DE QUELQUE MANIÈRE QUE CE SOIT NE DÉPASSERA LE MONTANT TOTAL DES DROITS DE LICENCE QUE VOUS AVEZ PAYÉS POUR LA LICENCE ACCORDÉE EN VERTU DU PRÉSENT CONTRAT, OU, SI AUCUN DROIT N'A ÉTÉ PAYÉ, LA SOMME D'UN DOLLAR. LES LIMITATIONS PRÉCÉDENTES S'APPLIQUERONT QUELLE QUE SOIT LA FORME DE TOUTE RÉCLAMATION AU TITRE DU PRÉSENT CONTRAT, QUE CE SOIT POUR VIOLATION OU RÉPUDIATION DE TOUTE AUTRE CONDITION DU PRÉSENT CONTRAT OU DE TOUT ÉCRIT Y AFFÉRENT, POUR NÉGLIGENCE, SUR LA BASE DE LA RESPONSABILITÉ STRICTE OU AUTRE. CECI EST VOTRE RECOURS EXCLUSIF. LES LIMITATIONS PRÉCÉDENTES S'APPLIQUERONT MÊME SI LE RECOURS SUSMENTIONNÉ NE REMPLIT PAS SON OBJECTIF ESSENTIEL. CERTAINES JURIDICTIONS N'AUTORISENT PAS L'EXCLUSION OU LA LIMITATION DE LA RESPONSABILITÉ TELLE QUE SPÉCIFIÉE ICI ET DANS LA MOINDRE MESURE PERMISE PAR LA LOI, CES EXCLUSIONS ET LIMITATIONS PEUVENT NE PAS VOUS ÊTRE APPLICABLES.

## 82.10 Durée et résiliation

### 82.10.1 Durée

La durée du présent accord commence à la date indiquée dans la confirmation de vente applicable et se poursuit jusqu'à sa résiliation conformément aux dispositions du présent accord.

### 82.10.2 Résiliation pour cause

Une partie peut mettre fin au présent accord (et à toutes les licences accordées en vertu des présentes) moyennant une notification écrite à l'autre partie dans le cas où cette dernière :

- Dépose ou fait déposer une demande de mise en faillite qui n'est pas rejetée dans les soixante (60) jours suivant le dépôt ou admet son incapacité à payer ses dettes à leur échéance, fait une cession au profit de ses créanciers ou cesse de fonctionner en tant qu'entreprise en activité ou de mener ses activités dans le cours normal des affaires, et cette cessation intervient immédiatement après notification ; ou
- Commet une violation substantielle de toute disposition du présent accord et ne remédie pas à cette violation dans les 30 jours (ou 10 jours après un défaut de paiement de toute taxe en vertu des présentes) suivant la réception de la notification de l'autre partie ou dans tout autre délai convenu entre les parties. En cas de résiliation motivée par la Société, le Client devra payer les frais impayés couvrant le reste de la durée de toutes les confirmations de vente après la date effective de résiliation. En aucun cas, une résiliation ne libère le Client de l'obligation de payer les frais payables à la Société pour la période précédant la date effective de la résiliation.

### 82.10.3 Résiliation pour raisons de commodité

Chaque partie a le droit de résilier toute licence pour des raisons de commodité, moyennant un préavis écrit d'au moins 10 jours à l'autre partie. Les frais payés d'avance ne sont pas remboursables.

### 82.10.4 Effets de la résiliation

À l'expiration ou à la résiliation du présent contrat, l'utilisation et l'accès du Client au logiciel et l'exécution par la Société de tous les services de support cesseront et tous les frais et autres montants dus à la Société seront immédiatement dus et payables par le Client. Dans les dix (10) jours suivant la date effective de la résiliation, chaque partie destinataire devra :

- Retourner à la partie émettrice ou, au choix de la partie émettrice, la partie réceptrice doit détruire tous les éléments d'information confidentielle alors en possession ou sous le contrôle de la partie réceptrice, y compris toute copie, tout extrait ou toute partie de ceux-ci ; et
- Sur demande, certifiée par écrit à la partie divulgateur qu'elle s'est conformée à ce qui précède.

## 82.11 Copyright

Le logiciel est protégé par les lois françaises sur le droit d'auteur et les dispositions des traités internationaux. Toute utilisation ou copie non autorisée du Logiciel est expressément interdite, sauf dans les cas expressément prévus par le présent Contrat. Toutes les copies que vous êtes autorisé à faire en vertu du présent Contrat doivent contenir les mêmes avis de droits d'auteur et de propriété non modifiés qui apparaissent dans le Logiciel.

## 82.12 Confidentialité

Vous reconnaissez et acceptez que le Logiciel et la documentation liée à son utilisation comprennent des informations propriétaires qui sont la propriété exclusive de la Société. Vous acceptez de ne pas utiliser ou divulguer ces informations propriétaires, sauf dans les cas autorisés par le présent contrat.

## 82.13 Intégralité de l'accord

Le présent accord constitue l'intégralité de l'accord entre les parties concernant l'objet des présentes et remplace toutes les négociations, propositions, représentations et accords antérieurs. Si une disposition du présent accord est jugée invalide, illégale ou inapplicable, la validité, la légalité ou l'applicabilité des autres dispositions ne seront en aucun cas affectées ou compromises. Aucune modification du présent accord n'est contraignante si elle n'est pas faite par écrit et acceptée par vous et la Société.

## 82.14 Composants logiciels tiers

Le logiciel utilise des composants open source, et votre utilisation du logiciel est soumise aux termes et conditions des licences logicielles régissant ces composants. Les composants open source du logiciel sont listés dans *Licences des composants externes utilisés dans WAPT*. Les informations sur les auteurs et la propriété des droits d'auteur de ces composants open source, ainsi que les conditions de licence applicables, peuvent être trouvées dans le répertoire d'installation du logiciel.

## 82.15 Droit applicable ; juridiction et lieu de juridiction

Dans toute réclamation découlant du présent accord ou s'y rapportant :

- La partie qui l'emporte en substance recouvrera ses frais et honoraires d'avocat raisonnables.
- La Société et vous-même acceptez la compétence du tribunal de Nantes (France) et renoncez à toute objection fondée sur la compétence ou le lieu de juridiction, y compris le forum non conveniens; toutefois, si la Société demande une mesure d'injonction, elle peut déposer une telle action là où, dans son jugement, la mesure peut être obtenue le plus efficacement.
- Les lois de la France s'appliqueront, indépendamment des dispositions relatives au choix de la loi applicable.

## 82.16 Divers

### 82.16.1 Notices

La Société peut aviser le Client par courrier électronique à l'adresse électronique du Client enregistrée auprès de la Société, ou par communication écrite envoyée par courrier affranchi de première classe ou par un service de livraison de nuit reconnu au niveau national à l'adresse du Client enregistrée auprès de la Société. Le Client peut notifier la Société par une communication écrite envoyée par courrier électronique à [commercial-tis@tranquil.it](mailto:commercial-tis@tranquil.it) ou par courrier affranchi en première classe ou par un service de livraison de nuit reconnu au niveau national, adressé à Tranquil IT, 12 avenue Jules Verne, 44230 Saint Sébastien sur Loire (France). La notification est réputée avoir été donnée à la réception ou, si elle est antérieure, deux (2) jours ouvrables après l'envoi, selon le cas. Toutes les communications et notifications à faire ou à donner en vertu du présent accord sont rédigées en anglais ou en français.

### 82.16.2 Clients du gouvernement américain

Si le Client est une entité du gouvernement fédéral des États-Unis, la Société fournit le logiciel, y compris le logiciel et la technologie associés, pour l'utilisation finale par le gouvernement fédéral uniquement en conformité avec ce qui suit : Les droits sur les données techniques du gouvernement comprennent uniquement les droits habituellement fournis au public avec un article ou un processus commercial et les droits sur les logiciels du gouvernement liés au logiciel comprennent uniquement les droits habituellement fournis au public, tels que définis dans le présent contrat. Les droits sur les données techniques et la licence de logiciel commercial habituel sont fournis conformément à FAR 12.211 (Données techniques) et FAR 12.212 (Logiciels) et, pour les transactions du ministère de la Défense, DFAR 252.227-7015 (Données techniques - Articles commerciaux) et DFAR 227.7202-3 (Droits sur les logiciels informatiques commerciaux ou la documentation des logiciels informatiques). Si des droits plus étendus sont nécessaires, un addendum écrit mutuellement acceptable, mentionnant spécifiquement ces droits, doit être inclus dans le présent accord.

### 82.16.3 Restrictions à l'export

Le Client reconnaît que le logiciel peut être soumis au contrôle des exportations et aux lois et règlements d'importation de :

- The United States, including without limitation the International Traffic in Arms Regulations (ITAR) (22 CFR Part 120-130), the National Industrial Security Program Operating Manual (NISPOM) (DoD 5 220.22-M), the U.S. Export Administration Regulations, 15 CFR Part 730-774, and other controls administered by the U.S. Department of Commerce, and the sanctions regulations administered by the U.S. Department of Treasury Office of Foreign Assets Control.
- L'Union européenne et ses États membres, y compris, sans limitation, le règlement (CE) n° 1334/2000 du Conseil.
- Et d'autres pays (collectivement, « loi sur l'exportation/importation »). Le Client s'engage à respecter toutes les lois d'exportation/importation applicables au logiciel et assume la responsabilité exclusive de l'obtention des licences et autres autorisations requises en vertu de la loi d'exportation/importation pour la livraison et l'utilisation du matériel logiciel. Le Licencié reconnaît et accepte que toute personne à qui le Logiciel est mis à disposition reconnaisse et accepte que le Logiciel ne soit pas exporté, réexporté ou transféré de toute autre manière vers des pays pour lesquels les États-Unis et/ou l'Union européenne maintiennent un embargo (collectivement, les « pays sous embargo »), ou un ressortissant ou résident de ces pays, ou vers toute personne

ou entité sur le territoire d'U. S. Liste des ressortissants spécialement désignés du département du Trésor des États-Unis ou la liste des parties ou entités refusées du département du Commerce des États-Unis, ou à toute personne figurant sur une liste comparable tenue par l'Union européenne ou ses États membres (collectivement, les « ressortissants désignés »). Les listes des pays sous embargo et des ressortissants désignés sont susceptibles d'être modifiées sans préavis. Le Client déclare et garantit que ni lui ni aucune des personnes à qui le logiciel est mis à disposition n'est situé dans un pays sous embargo ou un ressortissant désigné, ni ne réside dans un tel pays ou sous son contrôle. Le Client doit spécifiquement obtenir toutes les autorisations requises du gouvernement américain avant de transférer ou de divulguer de toute autre manière des données techniques ou des technologies (telles que ces termes sont définis dans 22 CFR Part 120.10 et 15 CFR Part 722, respectivement), à toute personne étrangère (telle que définie dans 22 CFR Part 120.16).

### 82.16.4 Généralités

Le Client ne doit pas céder ses droits en vertu des présentes, ni déléguer l'exécution de ses devoirs ou obligations en vertu des présentes, que ce soit par fusion, acquisition, vente d'actifs, application de la loi ou autrement, sans le consentement écrit préalable de la Société. Toute prétendue cession en violation de la phrase précédente est nulle et non avenue. Sous réserve de ce qui précède, le présent accord lie les successeurs et les ayants droit des parties et les assure au profit de ces dernières. À l'exception des affiliés du Client qui ont signé des confirmations de vente dans le cadre du présent accord, il n'y a pas de tiers bénéficiaire du présent accord. Sauf indication contraire dans le présent accord, le présent accord peut être modifié ou complété uniquement par un écrit qui fait explicitement référence au présent accord et qui est signé au nom des deux parties. Aucune renonciation ne sera implicite en ce qui concerne la conduite ou le défaut d'application des droits. Aucune renonciation ne sera effective si elle n'est pas formulée dans un écrit signé au nom de la partie contre laquelle la renonciation est invoquée. Si l'une des dispositions du présent accord est jugée invalide ou inapplicable, cette disposition sera appliquée dans toute la mesure permise par la loi et les autres dispositions resteront pleinement en vigueur. Les parties sont des entrepreneurs indépendants et rien de ce qui est contenu dans le présent accord ne doit être interprété comme créant une agence, un partenariat ou toute autre forme d'entreprise commune entre les parties. Le présent accord, y compris toutes les confirmations de vente applicables et les conditions séparées ou supplémentaires auxquelles il est fait référence dans le présent document, constituent l'intégralité de l'accord entre les parties concernant ce sujet et remplacent tous les accords, représentations, discussions, négociations et conventions antérieurs ou simultanés, qu'ils soient écrits ou oraux. À l'exception des obligations de paiement prévues par les présentes, aucune partie ne sera responsable envers l'autre partie ou un tiers pour un manquement ou un retard dans l'exécution de ses obligations en vertu du présent accord lorsque ce manquement ou ce retard est dû à une cause indépendante de la volonté de la partie concernée, y compris, sans limitation, les cas de force majeure, les ordonnances ou restrictions gouvernementales, l'incendie ou l'inondation, à condition que, dès la cessation de ces événements, cette partie exécute ou achève rapidement l'exécution de ses obligations en vertu des présentes.

---

## Licences des composants externes utilisés dans WAPT

---

Le développement du logiciel WAPT a commencé en mars 2012 ; il est porté en très grande partie par l'équipe de Tranquil IT. Avec WAPT >= 1.9, les développements réalisés dans le cadre de WAPT sont soumis à une *licence propriétaire*.

TABLEAU 1 – Licences des composants externes utilisés dans WAPT

| Composant WAPT                       | Licence                           |
|--------------------------------------|-----------------------------------|
| <b>Python</b>                        | Python Software License           |
| <b>Librairies Python</b>             | Licenses OpenSource diverses      |
| <b>Lazarus</b>                       | GNU Public Licence                |
| <b>Composants Lazarus</b>            | GNU Lesser General Public License |
| <b>Librairies Lazarus</b>            | Licenses OpenSource diverses      |
| <b>OpenSSL</b>                       | Openssl License                   |
| <b>Redistr. Microsoft Visual C++</b> | Microsoft Software License Terms  |
| <b>PostgreSQL</b>                    | PostgreSQL License                |
| <b>NSSM</b>                          | Public Domain                     |
| <b>Nginx</b>                         | 2-clause BSD-like license         |