WAPT Documentation

Version 2.0.0

Tranquil-IT Systems

janv. 09, 2024

Présentation de WAPT

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 :

- Install software and configurations silently.
- Maintain up to date an installed base of software and configurations.
- Configure software at the system and user level to reduce the load on support teams.
- Remove unwanted or out of cycle software and configurations silently.
- Reduce your need for support by your IT teams, whose reaction times are often long because of their workloads.
- Reduce as much as possible the consumption of bandwidth on remote sites to preserve it for productive uses.

Pour les RSSI

- Pilot the software installed base to converge to a security standard acceptable to the organization.
- Prepare your enterprise for the coming GDPR and help your DPO keep his register of data processing, because you two will become close colleagues.
- No longer tolerate machines operating in *Administrator* mode.
- No longer tolerate users downloading and running software binaries from their home directory.
- Start applying SRPs (Software Restriction Policies), also known as *Applocker* or WDAC (Windows Defender Application Control) to improve application level IT security.
- Reduce the level of exposure to software vulnerabilities and lateral movement attacks.
- Bring up audit indicators for a better knowledge of the state of installed IT devices and their global security level.
- Be prompt to deploy updates to react to cyber attacks like Wannacry or notPetya.

Pour les utilisateurs finaux

- Have installed software configured to work well in the context of your Organization and trust that they will work correctly.
- Give *Users* more autonomy to install software safely and reliably.
- Have better working and more predictable professional systems because of standard software configurations.

CHAPITRE 1

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 Elémentaire 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 :

- Common ghosting methods (*Clonezilla* or *Ghost*) are efficient on homogeneous IT infrastructures with roaming user profiles.
- Deployment tools (*OCSInventory* or *WPKG*) can broadcast software but do not easily allow software level or user level customizations that are useful to prevent or limit user support requests.
- Software from smaller vendors often need *Local Administrator* rights to run properly.
- Currently available solutions to address theses problems are either too expensive or too inefficient, and they are in every case too complex.

1.3.2 WAPT development hypotheses and motivations

Le développement de WAPT est animé par deux principes :

- What is **complicated** should be made **simple**.
- What is **simple** should be made **trivial**.

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

- Sysadmins should know a scripting language and WAPT has chosen Python for the depth and breadth of its libraries.
- Sysadmins who have little experience with scripting languages must find inspiration in simple and efficient examples that they'll adapt to fit their needs.
- Sysadmins must be able to communicate on the efficiency of their actions to their superiors and report process gaps to internal or external auditors.
- Sysadmins must be able to collaborate with their IT team ; thereby WAPT local repositories provide signed packages that they can trust to be deployed on their network. Alternatively, they can choose external public repositories providing them the security guarantees that they consider sufficient.
- Sysadmins are aware that user workstations serve business purposes and some customizations must be possible. The adaptation of the infrastructure to the business needs is facilitated by the notion of groups and OU (Organizational Units) ; it allows to select a large number of machines to customize their configuration.

CHAPITRE 2

Principes fondamentaux

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 :

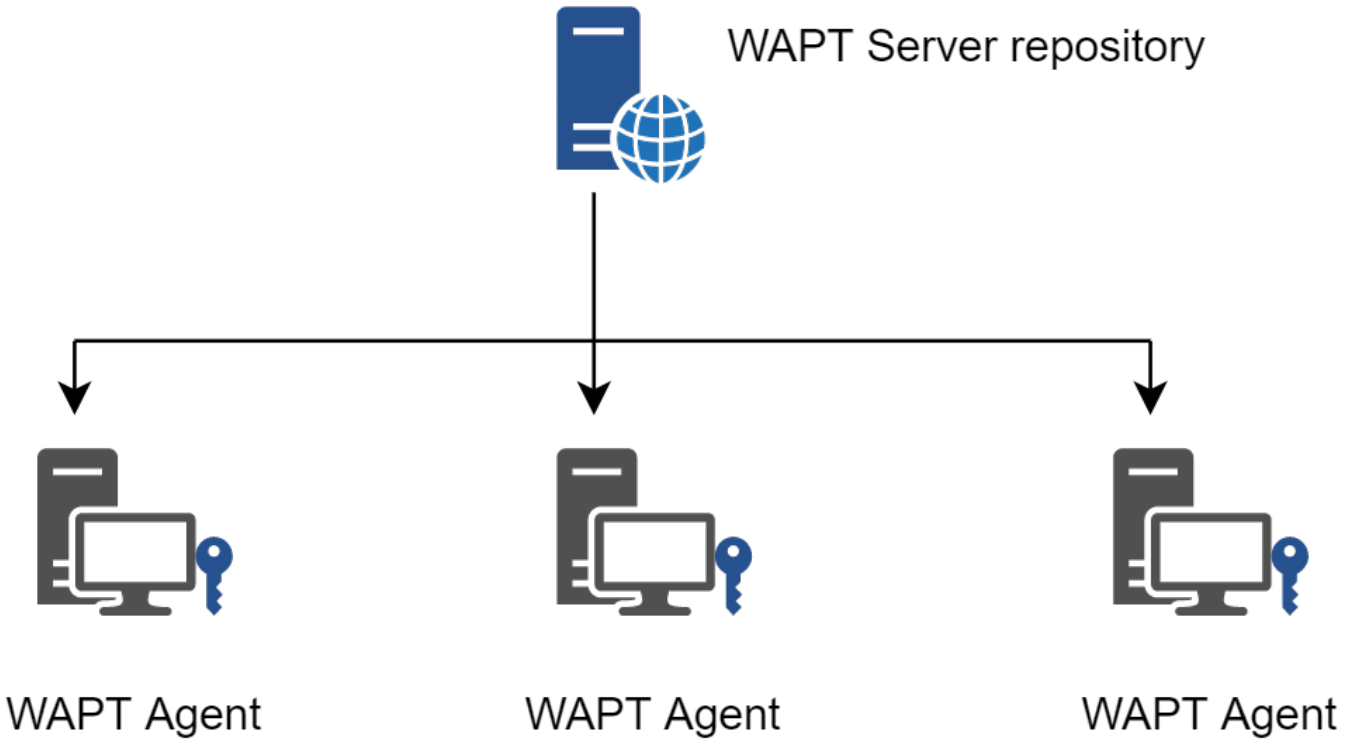


FIG. 1 – Réplication et dépots multiples

- A small form factor and no maintenance appliance with the role of secondary repository is deployed on the local network of each remote site ; a workstation can also be used, although it may not be up and running if you want to connect to it.
- The remote repository replicates the packages from the main repository.
- The WAPT clients connect in priority with the repository that is the closest to them, the local repository.

Pour en savoir plus sur le dépôt répliqué, consultez la documentation sur *Réplication d'un dépot*.

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.

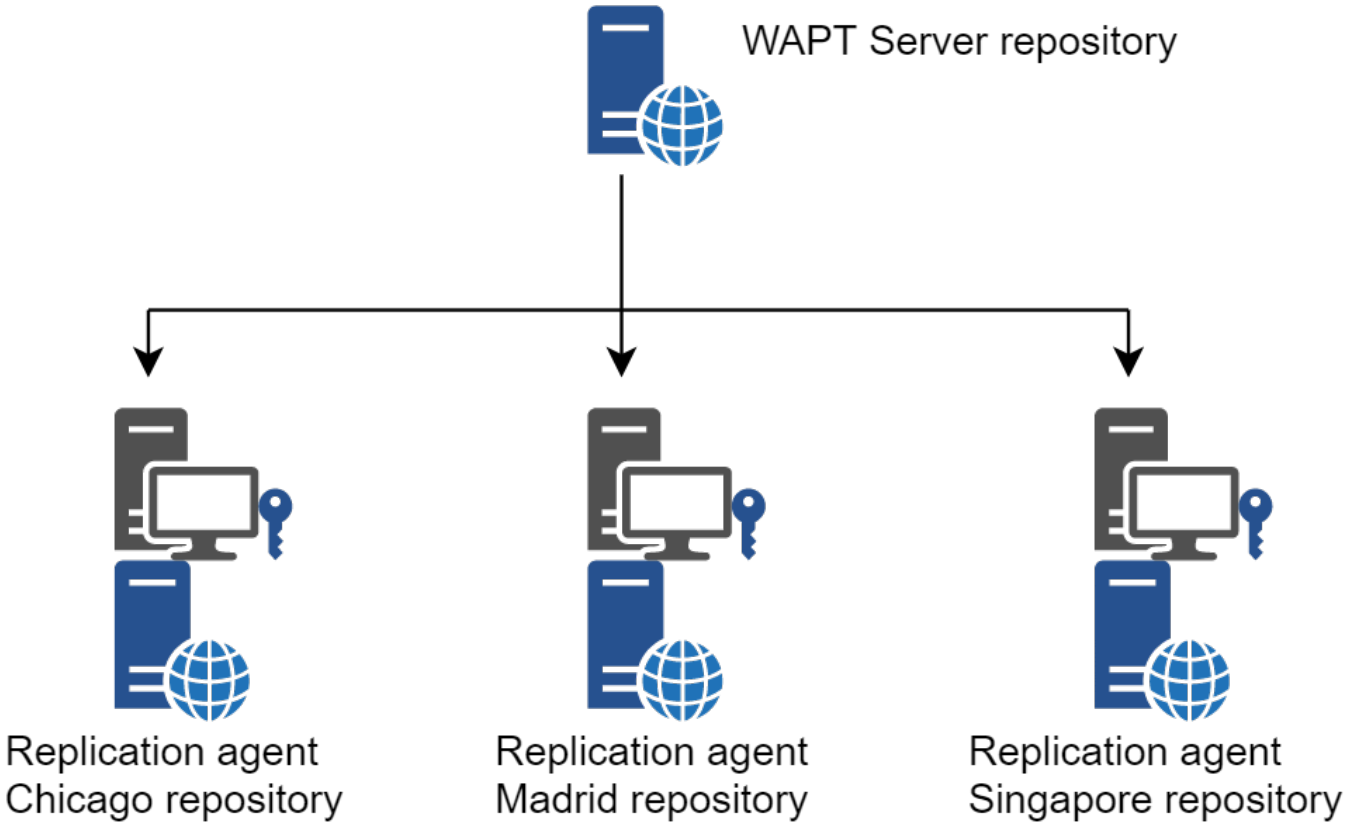


FIG. 2 – Réplication des dépôts WAPT

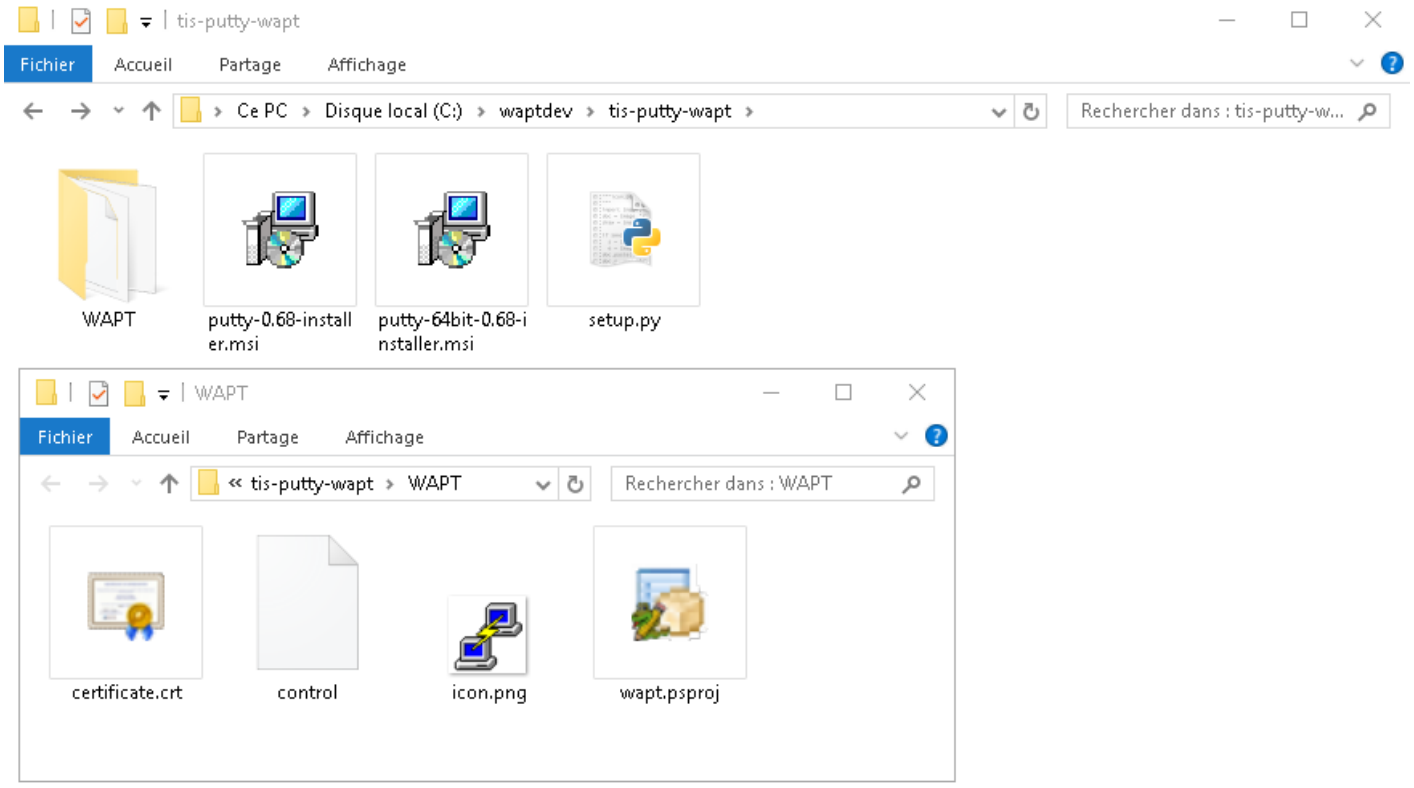


FIG. 3 – Structure d'un paquet WAPT

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.

Each group often correspond to :
- a service in an organization (ex : **accounting**).
- une pièce, un bâtiment, etc.

Indication : A host can be a member of several groups.

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, exemple : **OU=piece1,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.

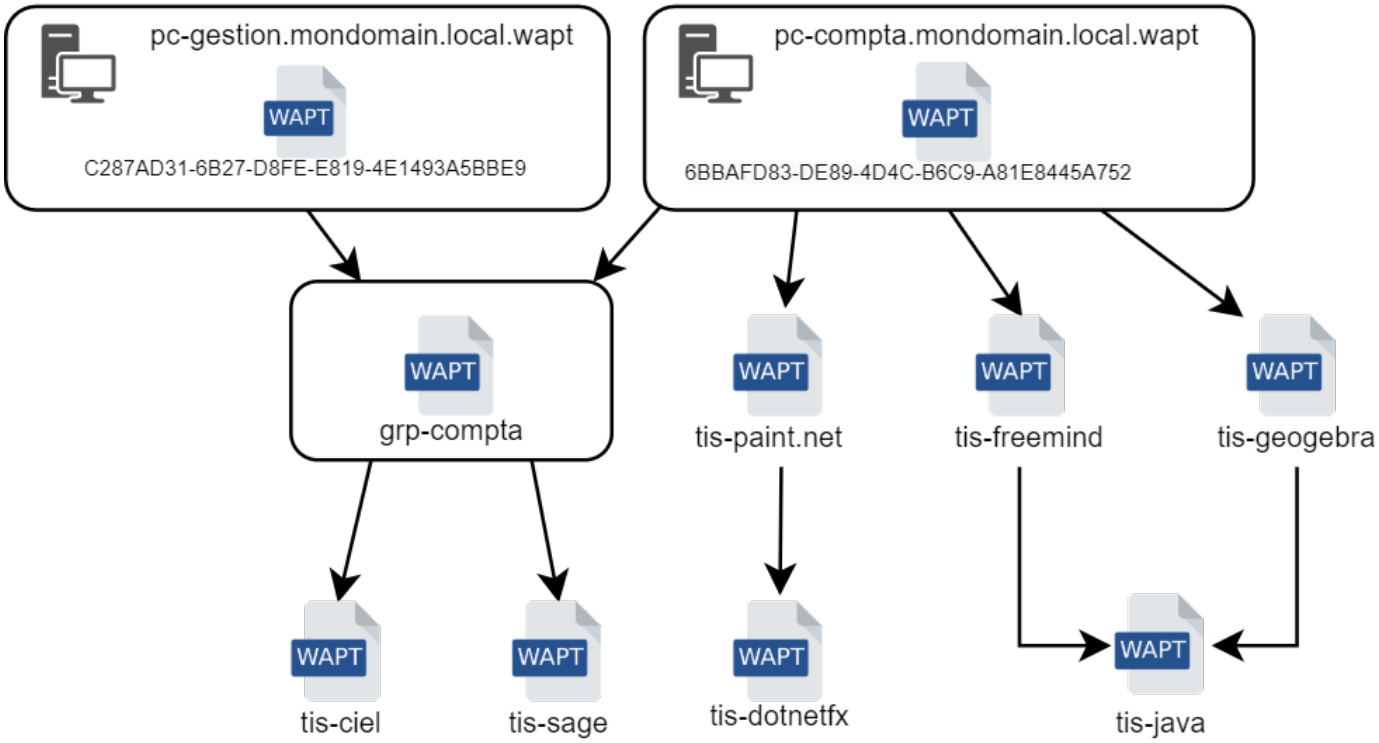


FIG. 5 – Schéma conceptuel du mécanisme de dépendance

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 hsah 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.

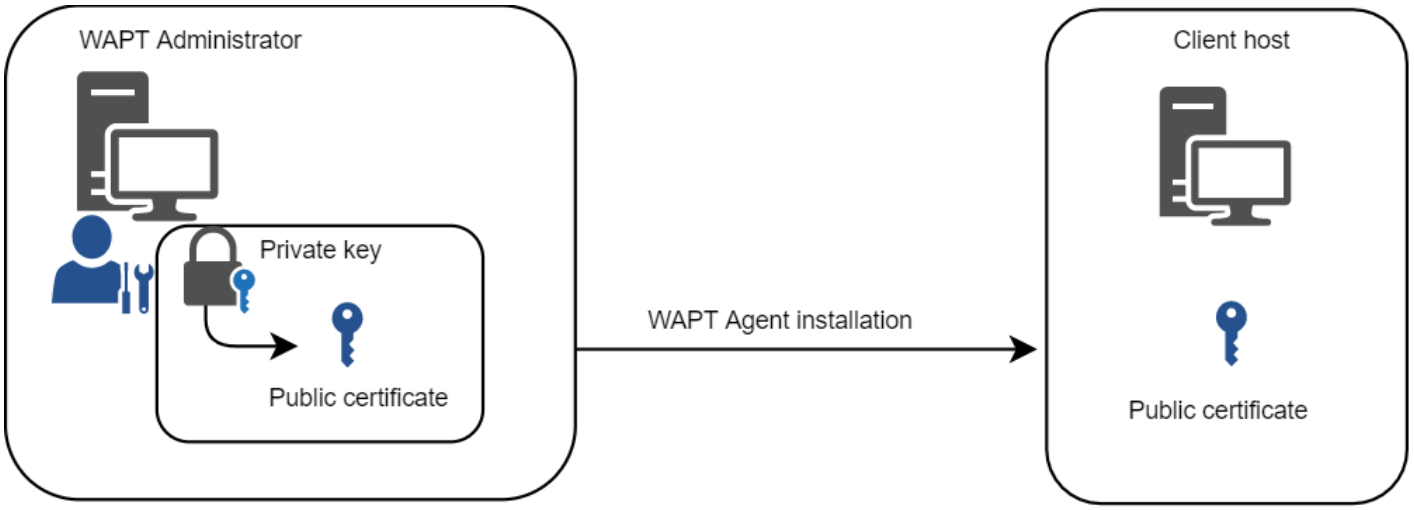


Fig. 6 – Clé privée / certificat public

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 situées 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 :
- A PKI (Public Key Infrastructure).
- ACLs (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 :
- Code-signing to allow IT admins to package, sign and deploy WAPT packages containing executable loads (i.e. `setup.py`).
- CA to delegate to other IT admins the right to issue certificates.
- No right to limit IT admins to only deploying packages containing non executable loads (i.e. configure hosts).

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 (Access Control Lists).

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*.

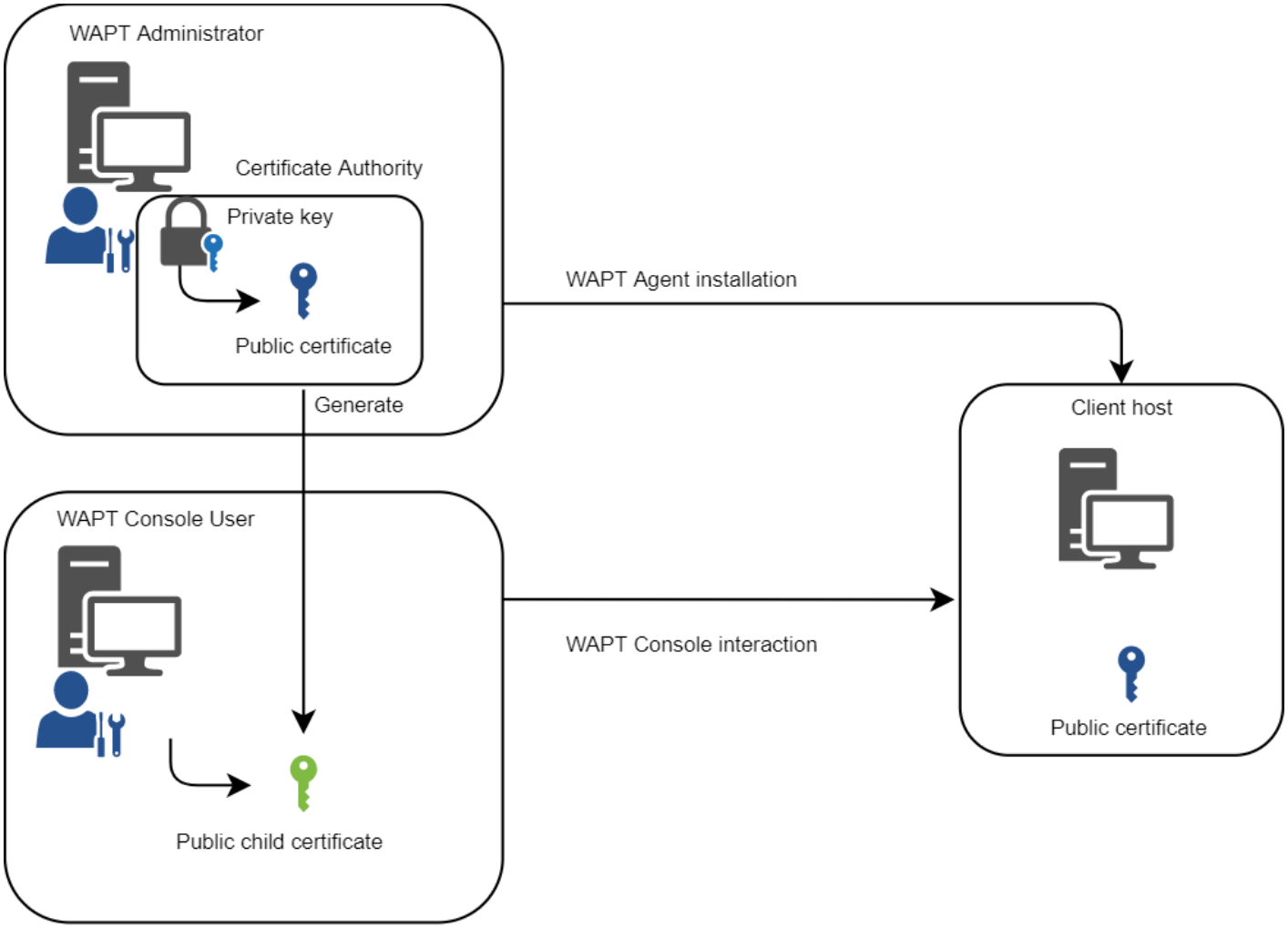


FIG. 7 – Différenciation des rôles des utilisateurs WAPT

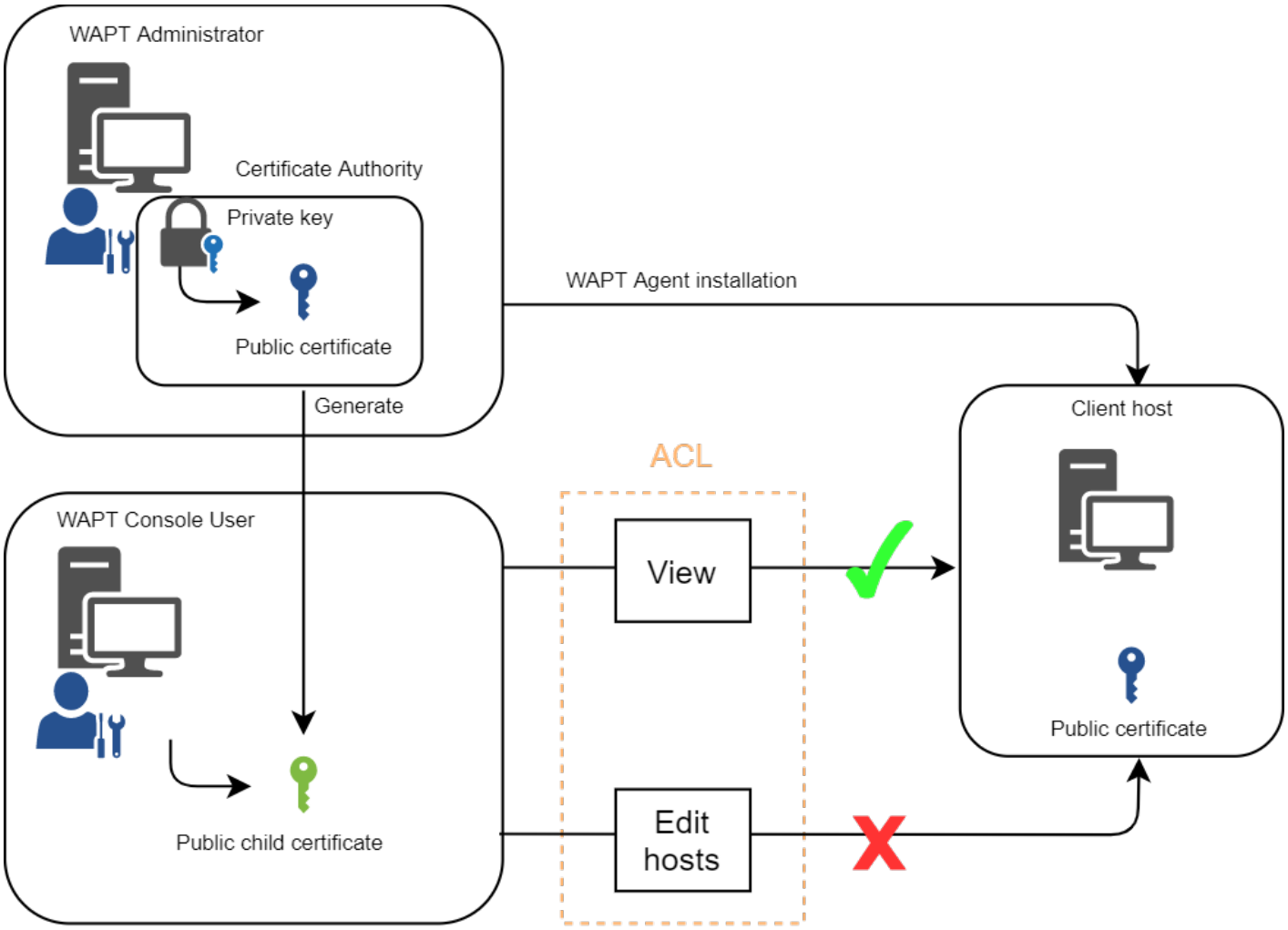


FIG. 8 – Différenciation des rôles ACL

CHAPITRE 3

Mode de fonctionnement WAPT

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.

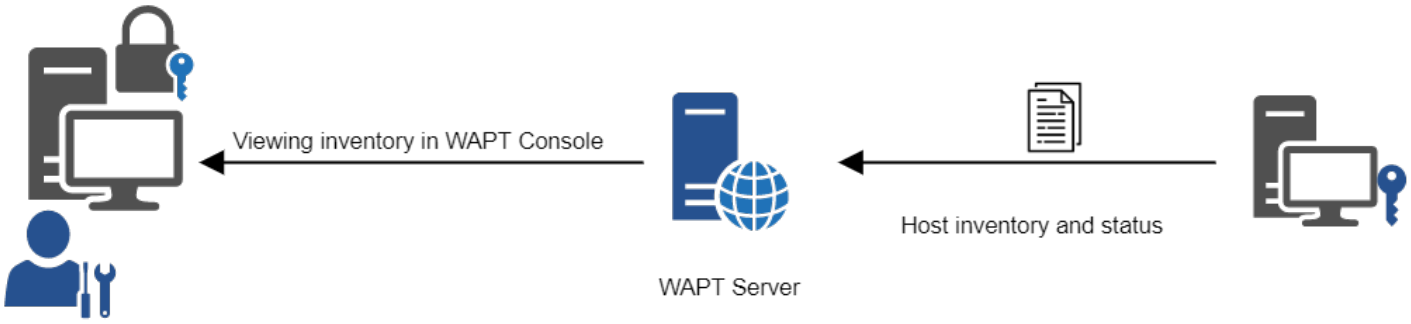


Fig. 1 – Fonctionnement de la remontée d'inventaire

— When first registering with the WAPT Server, the WAPT agent sends the entire inventory (BIOS, hardware, software) to the server.
— When the WAPT agent updates, the WAPT agent will report its inventory status to the WAPT Server.

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

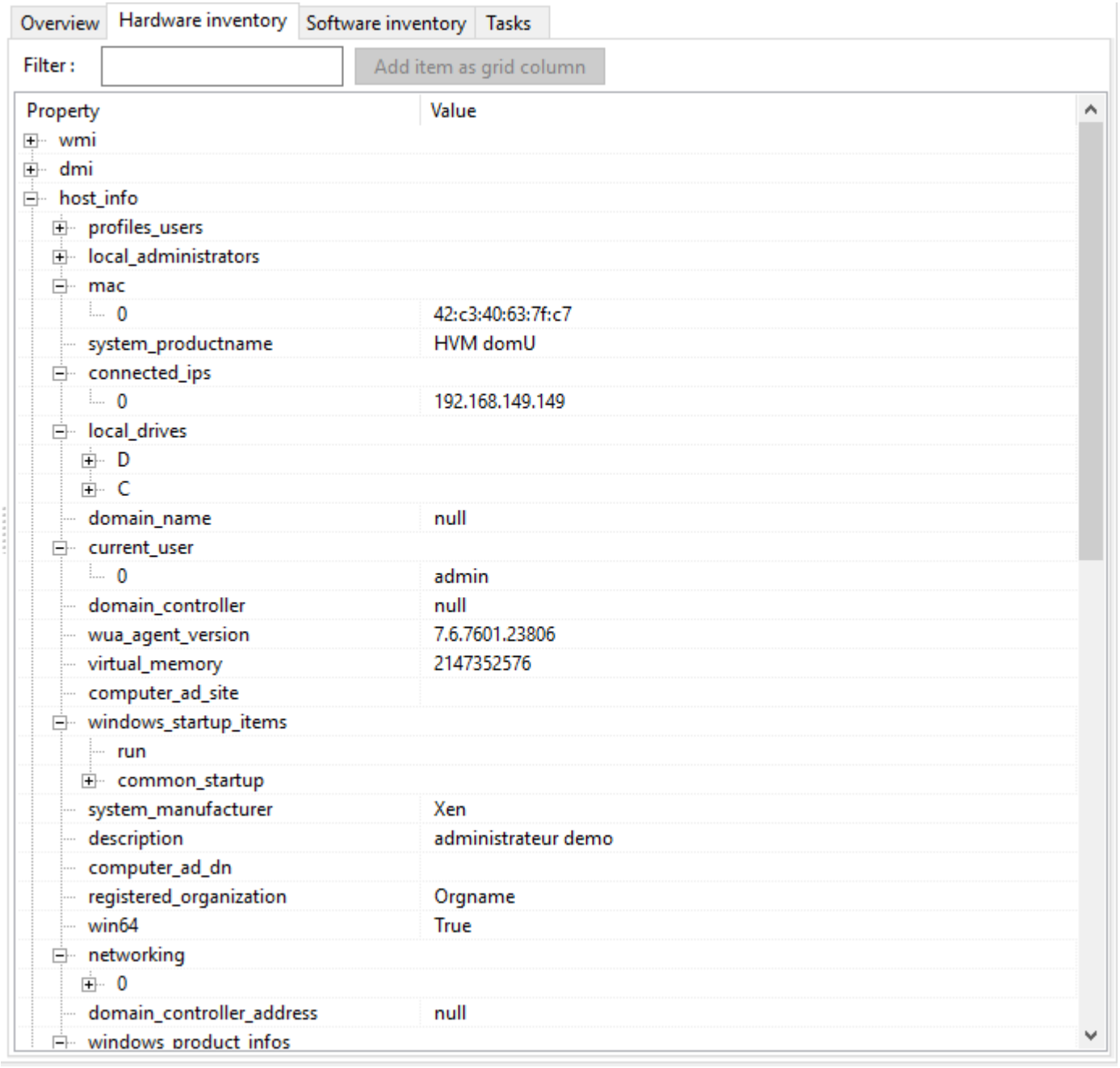


FIG. 2 – L'inventaire dans la console WAPT

3.2 La remontée des informations d'inventaire

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

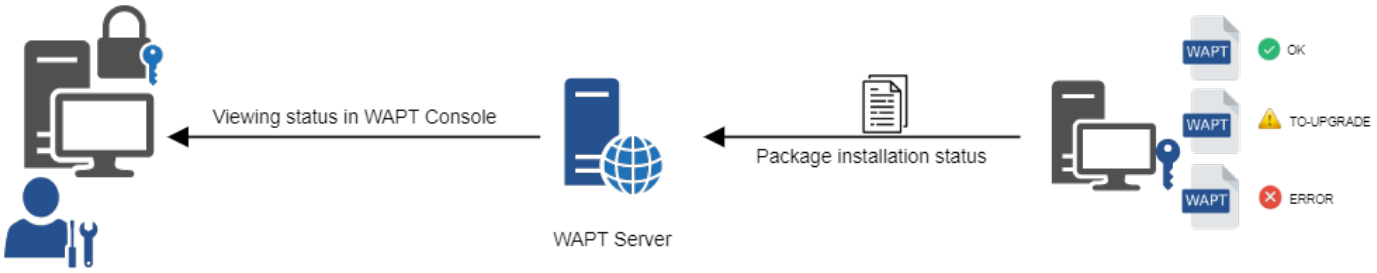


Fig. 3 – La remontée du statut des paquets vers le serveur WAPT

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.

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 dépot 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 :
- To install a **base**, a **group** or a **unit** package if it is available.
- To remove obsolete packages.
- To resolve package dependencies and conflicts.
- To make sure all installed WAPT packages are up to date compared to the ones stored on the repository.
- To regularly update the WAPT server with its hardware status and the status of installed software.

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 :
- On triggerring an **update**, the agent downloads *NEED-UPGRADE* or *NEED-INSTALL* packages and stores them in the cache folder.
- On triggerring an **upgrade**, the agent unzips the packages into a temporary folder.
- The `setup.py` content is parsed and stored in WAPT agent database located in `C:\Program Files (x86)\wapt\db\waptdb.sqlite`.
- The file `setup.py` is executed and the software is installed from unzipped files.
- In case of success : the downloaded packages and unzipped files are deleted. An **OK** status is returned to the WAPT Server.
- In case of failure : the downloaded packages are kept and the unzipped files are deleted. An **ERROR** status is returned to the WAPT Server.

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 :
- The `setup.py` content is retrieved from WAPT agent database located in `C:\Program Files (x86)\wapt\db\waptdb.sqlite`.
- The WAPT agent looks up the `UninstallString` in the local database.
- If defined in the `setup.py` copied into the local database during initial installation of the WAPT package, the **uninstall()** function is executed.

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

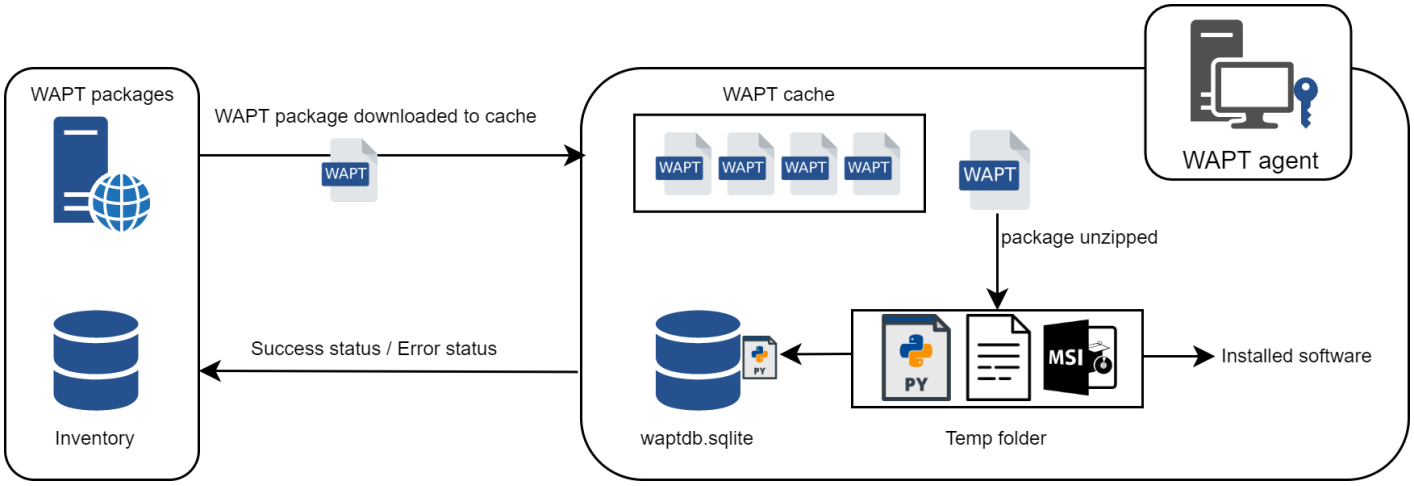


FIG. 5 – Comportement d'installation de WAPT

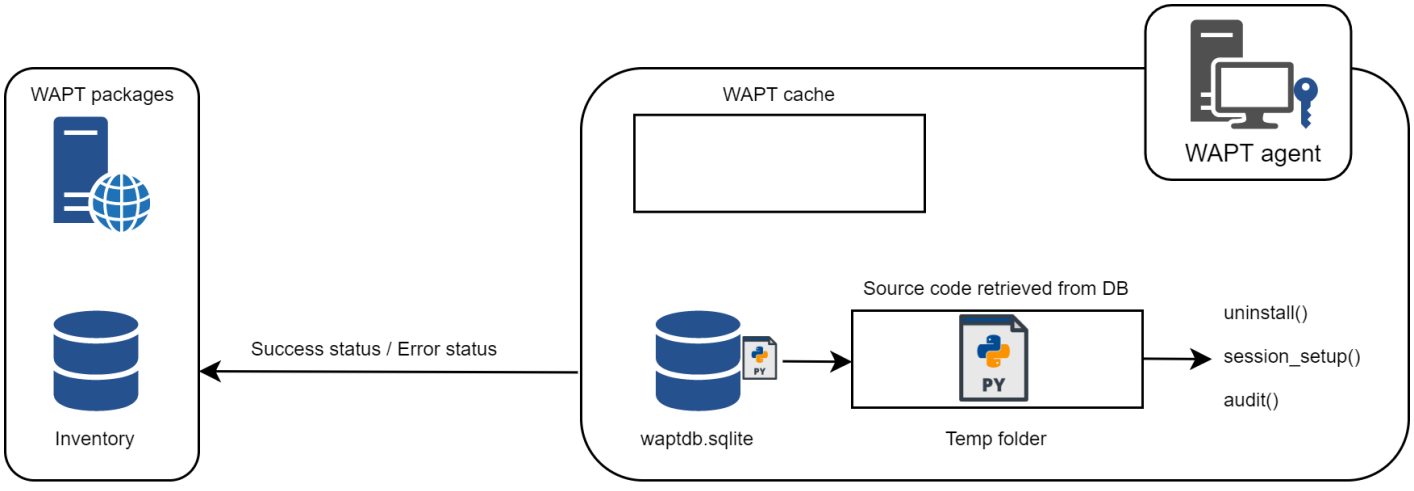


FIG. 6 – Comportement de l'agent WAPT avec la désinstallation, le session_setup et l'audit

3.5 Diagramme complet du fonctionnement de WAPT

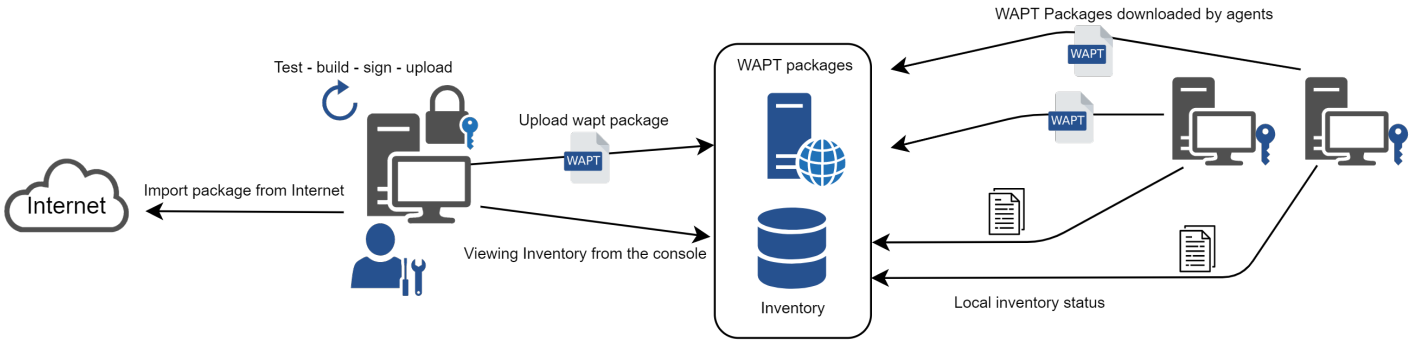


FIG. 7 – Mode de fonctionnement général du WAPT

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 :

- Import packages from an external repository (or create a new package from scratch).
- Test, validate, build and then sign the package.
- Upload the package onto the main repository.
- Packages are automatically downloaded by WAPT clients.
- Packages are executed based on the selected method :
  - The *Administrator* forces the **`upgrade`**.
  - The *Administrator* proposes the **`upgrade`** at *User*.
  - A scheduled task launches the upgrade.
  - The upgrade is executed when the machine shuts down.
  - The *User* chooses the right time for herself (at shutdown or using the *self-service*).
- Inventory information feedback.
- The updated inventory is reported in the WAPT console.

CHAPITRE 4

Architecture du serveur WAPT

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

— The *repository role* for distributing packages.
— The *inventory* and *central server* role for hardware and software inventory.
— The *proxy* role to relay actions between the WAPT console and the WAPT agents.

4.1 Fonctionnement du dépôt WAPT

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

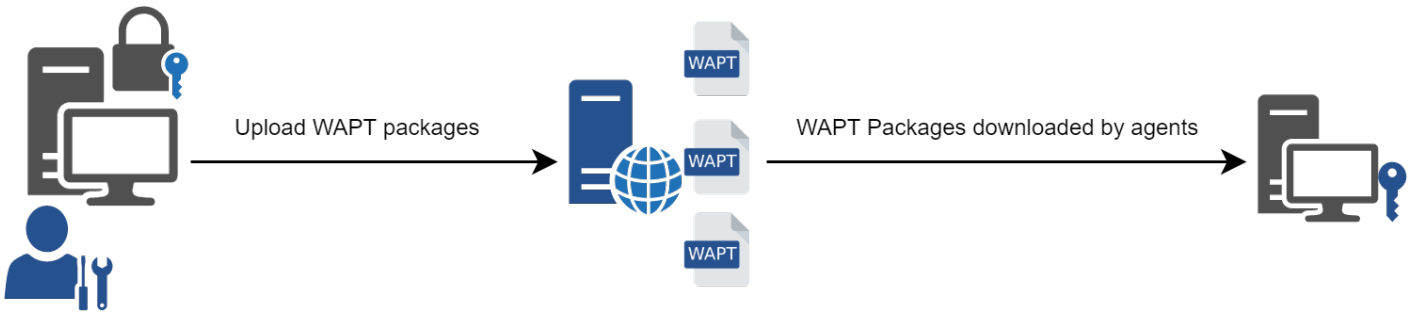


FIG. 1 – Fonctionnement du dépôt WAPT

— The repository role is accomplished by a **`Nginx`** web server.
— The repository allows the distribution of WAPT packages, the installers for *waptagent* and *waptsetup*.
— WAPT packages are available via a web browser by visiting https://srvwapt.mydomain.lan/wapt.
— The *host* packages are stored in a directory that is not accessible by default (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 :

- Hardware inventory.
- Software inventory.
- WAPT packages status.
- Tasks status (*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.

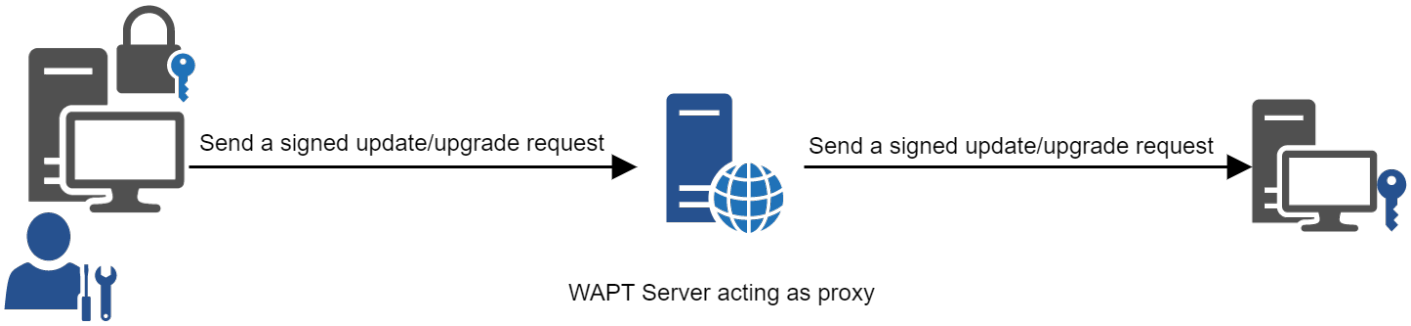


Fig. 2 – Fonctionnement du proxy WAPT

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 *signing actions relayées aux agents WAPT*.

CHAPITRE 5

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 :
- Doing conditional loops (if … then … else …).
- Copying, pasting, moving files and directories.
- Checking whether files or directories exist.
- Checking whether registry keys exist.
- Checking access rights, modifying access rights.
- Looking up information on external data sources (LDAP, databases, files, etc).
- And more.

5.2 La puissance de WAPT

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

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

- **What was complicated is made simple**.
- **What was simple is made trivial**.

CHAPITRE 6

Comparaison des caractéristiques des versions de WAPT

6.1 Résumé des principes de fonctionnement de WAPT

- **WAPT is agent based to allow no inbound open port** in host's firewalls that initiate a secured bi-directional websocket with the server for allowing real-time reporting and actions.
- WAPT works with Trusted Data Gateways using simple task scheduling.
- WAPT works on the principle of smoothly pulling updates and then applying upgrades at a convenient time (works with low / intermittent bandwidth, high latency, high jitter networks).
- WAPT does not require an Active Directory to work (works with Windows Home edition too) ; however, WAPT will show the host in its Active Directory tree if the host is joined to an AD.
- Methods for deploying WAPT agent :
  1. Using a GPO (Group Policy Object) or an Ansible script.
  2. Manually after having downloaded the agent from the WAPT server or using SSH (Secured Shell).
- Methods for registering hosts with the WAPT server :
  1. Automatically using the host's kerberos account.
  2. Manually with the WAPT *Superadmin* login and password.
- Upgrades may be triggered :
  1. Upon shutdown of the host, this is the standard mode.
  2. By an authorized WAPT Administrator in an emergency (ex : patching critical vulnerabilities running in the wild).
  3. By the user herself at a time she chooses (ex : 24/7 nursing cart unused during breaks with a simple click).
  4. Via a scheduled task running at a predetermined time (best for servers).
- Security is insured with :
  1. Signing of WAPT packages using asymmetric cryptography.
  2. Authentication of hosts against the WAPT server using symmetric cryptography on registering.
  3. Confidentiality of the WAPT server using WAPT deployed client certificates.
  4. Using of ACL to define what an administrator is allowed to view or what actions he is allowed to perform according to his certificate.

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 *friednly 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
Déploiement, mise à jour et suppression des logiciels sur les hôtes	✅	✅	✅[1]
Maintenance et support (voir note de bas de page pour les conditions)	Équipe Tranquil IT[4]	Forum Tranquil IT[?]	Communauté Open-Source
Licensed under	Proprietary	Proprietary	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 **configurations dans le contexte du SYSTÈME**	✅	✅	✅[?]
Déployer et mettre à jour les **configurations dans le contexte de l'UTILISATEUR**	✅	✅	✅[?]
Obtenez un **inventaire complet** du matériel, des logiciels et des paquets WAPT appliqués	✅	✅	✅
Bénéficier du **seilf-service différencié** (les utilisateurs autorisés peuvent installer les logiciels autorisés à partir du store de paquets WAPT autorisés)	✅	❌	❌
Bénéficiez de **Mises à jour Windows simplifiées** 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 decondaires WAPT **pour préserver la bande passante** pour les scénarios *Edge Computing*	✅	❌	❌
Accédez à des **paquets WAPT prêts à être déployés** pour des logiciels communs gratuits	✅	✅	✅[?]
Travailler avec des **recettes python facilement vérifiables** pour l'installation, la mise à jour et la suppression de logiciels et de configurations	✅	✅	✅[?]
Bénéficiez de **centaines d'assistants** pour simplifier le conditionnement des logiciels	✅[2]	✅	✅[?]
Chiffrez vos données sensibles 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 **facile, automatisée et toujours à jour**	✅	❌	❌
Profitez de la puissance de SQL intégrée à la console WAPT pour créer les **rapports dont vous avez besoin pour votre travail quotidien d'administrateur système ou dont votre organisation a besoin pour prendre des décisions budgétaires**	✅	❌	❌
Authentifiez vos *Administrateurs* WAPT avec **Active Directory, LDAP, ou avec leurs certificats personnels**	✅	❌	❌[3]
Bénéficiez de rôles différenciés entre vos *Développeurs de Paquets* et vos *Déployeurs de Paquet* afin que vous puissiez **déléguer vos pouvoirs WAPT aux personnes les plus adéquates** (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 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 et de confidentialité que WAPT (nécessite un logiciel supplémentaire).	✅	❌	❌
Poursuite de la prise en charge de Windows XP dans WAPT pour les machines-outils d'usine, les équipements médicaux des hôpitaux, les instruments de recherche coûteux et difficiles à remplacer, etc	✅[5]	❌	❌
Mise à jour des paquets directement dans la console WAPT avec la fonction `update_package`	✅	❌	❌

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	Enterprise	Discovery
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 www.virustotal.com	✓	✓[7]
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**
of WAPT package developers, package deployers, self-service users and for the initial registering of the WAPT agents with the WAPT Server. In addition, the display of WAPT equipped devices in the WAPT console follow the same structure as the hierarchical structure of the Organization's Active Directory OU.

— **Role separation between package developers and package deployers**.
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 *déployeurs* de paquets) et avec des rigths 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.

— **Differentiated self-service**.
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.

— **Advanced reporting for corporate teams**.
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.

— **Dynamic repository configuration**.
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.
The replication role is deployed through a WAPT package that enables the **`Nginx web server`** and configures scheduling, packages types, packages sync, and much more.
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.

— **Integration with 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 :

— That manage large installed bases of devices (generally above 300 units).
— That are spread geographically with many subsidiaries or production sites.
— That require a strong traceability of actions performed on the installed base of devices for reasons of audit or security.
— That value secured and proven solutions in their IT sourcing.

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

You may choose to train your IT team on any particularity of 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.

CHAPITRE 7

Préconisations 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
De 0 a 200 postes	2 CPU	2024 Mio
A partir de 200 postes	4 CPU	4096 Mio

- 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). Oui, WAPT peut être très rapide !

7.2 Préconisations matérielles

WAPT server are available on Linux and Windows :
— Pour Linux, **Debian 10**, **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**.
— 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 (1903 ou plus).
The WAPT Server will only run on **64bit** based system.

CHAPITRE 8

Conseils avant l'installation

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.

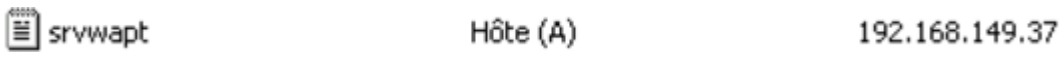


FIG. 1 – Configuring the A field in Windows RSAT

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*.

CHAPITRE 9

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 Configuring the network parameters

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
```

— Check whether the NTP service is installed, started and whether the time is correct.

```
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
```

— 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.

— Install `apt-transport-https` for the use of https.

```
```

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

— Retrieve the *.gpg* key and add Tranquil IT's repository.

```
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
```

— Create the `wapt.conf` file in `/etc/apt/auth.conf.d` to store your login information.

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
```

— Apply the correct ACLs on `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
unset DEBIAN_FRONTEND
```

9.4 Post-Configuration

Attention : Pour que la post-configuration fonctionne correctement, vous devez d'abord avoir correctement configuré le *nom d'hôte* du serveur WAPT. Pour vérifier, utilisez la commande **`echo $(hostname)`** qui doit retourner l'adresse DNS qui sera utilisée 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 faudra écrasé la configuration après le 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 qu'utilisateur **root**.

— Lancer le script.

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

— Cliquez sur *Yes* pour exécuter le script postconf.

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

     < yes >          < no >
```

— Choisissez un mot de passe 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 >
```

— 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 n° 1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
  — Le choix n°2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
  — Le choix n°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
(x) 3 Disable kerberos but registration require strong authentication
-------------------------------------------------------------------------
                    < OK >          < Cancel >
```

— Sélectionnez *OK* pour démarrer le serveur WAPT.

```
Press OK to start waptserver

      < OK >
```

— Sélectionnez *Yes* pour configurer Nginx.

```
Do you want to configure nginx?

    < Yes >        < No >
```

— Indiquez 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, sélectionner *OK* pour redémarrer **Nginx** :

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

      < OK >
```

La post-configuration est maintenant terminée.

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

                < OK >
```

Détail des arguments possibles du script de post-configuration :

Options	Description
`--force-https`	Configures **Nginx** so that *port 80 is permanently redirected to 443*

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

CHAPITRE 10

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

— Configure the IP address of the WAPT Server in `/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
```

— Check whether the NTP service is installed, started and whether the time is correct.

```
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
```

— 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.

— Install `apt-transport-https` for the use of https.

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

— Retrieve the *.gpg* key and add Tranquil IT's repository.

```
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
```

— Create `wapt.conf` in `/etc/apt/auth.conf.d` to store your login information.

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
```

— Apply the correct ACLs on `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
unset DEBIAN_FRONTEND
```

10.4 Post-configuration

Attention : Pour que la post-configuration fonctionne correctement, vous devez d'abord avoir correctement configuré le *nom d'hôte* du serveur WAPT. Pour vérifier, utilisez la commande **`echo $(hostname)`** qui doit retourner l'adresse DNS qui sera utilisée 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 faudra écrasé la configuration après le 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 qu'utilisateur **root**.

— Lancer le script.

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

— Cliquez sur *Yes* pour exécuter le script postconf.

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

     < yes >          < no >
```

— Choisissez un mot de passe 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 >
```

— 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 n° 1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
  — Le choix n°2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
  — Le choix n°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
(x) 3 Disable kerberos but registration require strong authentication
--------------------------------------------------------------------------
                    < OK >          < Cancel >
```

— Sélectionnez *OK* pour démarrer le serveur WAPT.

```
Press OK to start waptserver

      < OK >
```

— Sélectionnez *Yes* pour configurer Nginx.

```
Do you want to configure nginx?

    < Yes >        < No >
```

— Indiquez 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, sélectionner *OK* pour redémarrer **Nginx** :

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

      < OK >
```

La post-configuration est maintenant terminée.

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

                < OK >
```

Détail des arguments possibles du script de post-configuration :

Options	Description
`--force-https`	Configures **Nginx** so that *port 80 is permanently redirected to 443*

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

CHAPITRE 11

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.

— Adding Tranquil IT's repository.

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-2.0/
username=user
password=password
enabled=1
gpgcheck=1
EOF
```

11.3 Installer des paquets complémentaires

— Retrieving the key `.gpg`.

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

— Install all necessary packages.

```
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 correctement configuré le *nom d'hôte* du serveur WAPT. Pour vérifier, utilisez la commande **`echo $(hostname)`** qui doit retourner l'adresse DNS qui sera utilisée 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 faudra écrasé la configuration après le 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 qu'utilisateur **root**.

— Lancer le script.

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

— Cliquer sur *Oui* pour lancer le script postconf.

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

    < yes >         < no >
```

— Choisissez un mot de passe 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 >
```

— 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 n° 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 pourrez l'activer aussi plus tard).
  — Le choix n°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
(x) 3 Disable kerberos but registration require strong authentication
------------------------------------------------------------------------------
                     < OK >          < Cancel >
```

— Sélectionner *OK* pour démarrer le serveur WAPT.

```
Press OK to start waptserver

      < OK >
```

— Sélectionner *OK* pour configurer Nginx.

```
Do you want to configure nginx?

    < Yes >        < No >
```

— Indiquez 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, sélectionner *OK* pour redémarrer **Nginx** :

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

      < OK >
```

La post-configuration est maintenant terminée.

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

                < OK >
```

Détail des arguments possibles du script de post-configuration :

Options	Description
`--force-https`	Configures **Nginx** so that *port 80 is permanently redirected to 443*

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

CHAPITRE 12

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\nginx\conf\nginx.conf C:\wapt\waptserver\nginx\conf\nginx.conf.old
```

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

```
copy C:\wapt\waptserver\nginx\conf\nginx.conf.old C:\wapt\waptserver\nginx\conf\nginx.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.
— Sautez la création de la clé personnelle, nous créerons *votre certificat* plus tard.
— Sauter la construction de l'agent WAPT, *nous le ferons plus tard*.
— Cliquez sur le bouton *Install* pour lancer l'installation, attendez que l'installation soit terminée.
— Cliquez sur *Terminé* pour fermer la fenêtre.

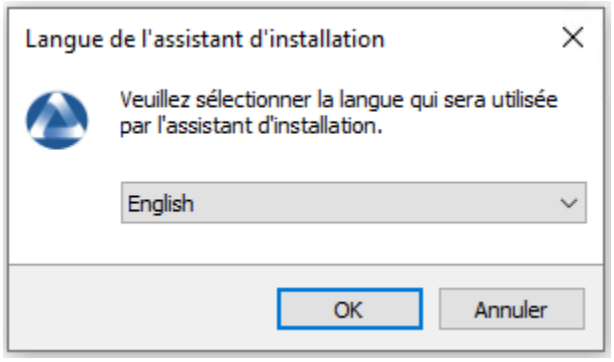


FIG. 1 – Choisir la langue pour WAPT

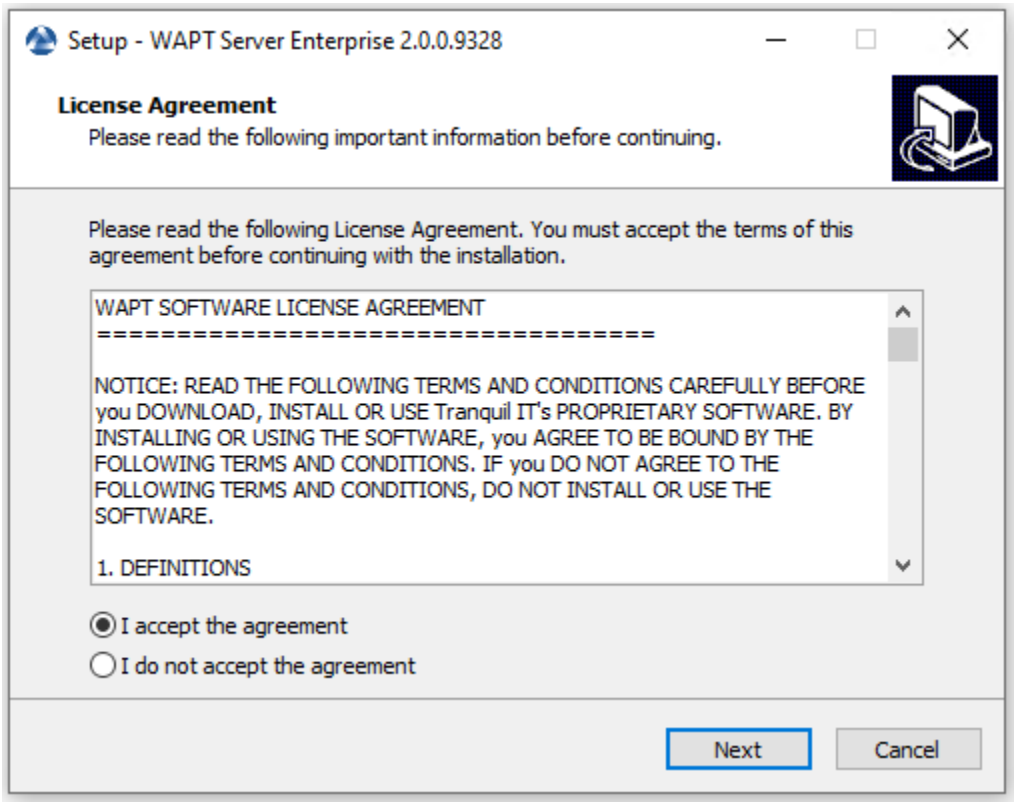


FIG. 2 – Accepter les conditions de la licence WAPT

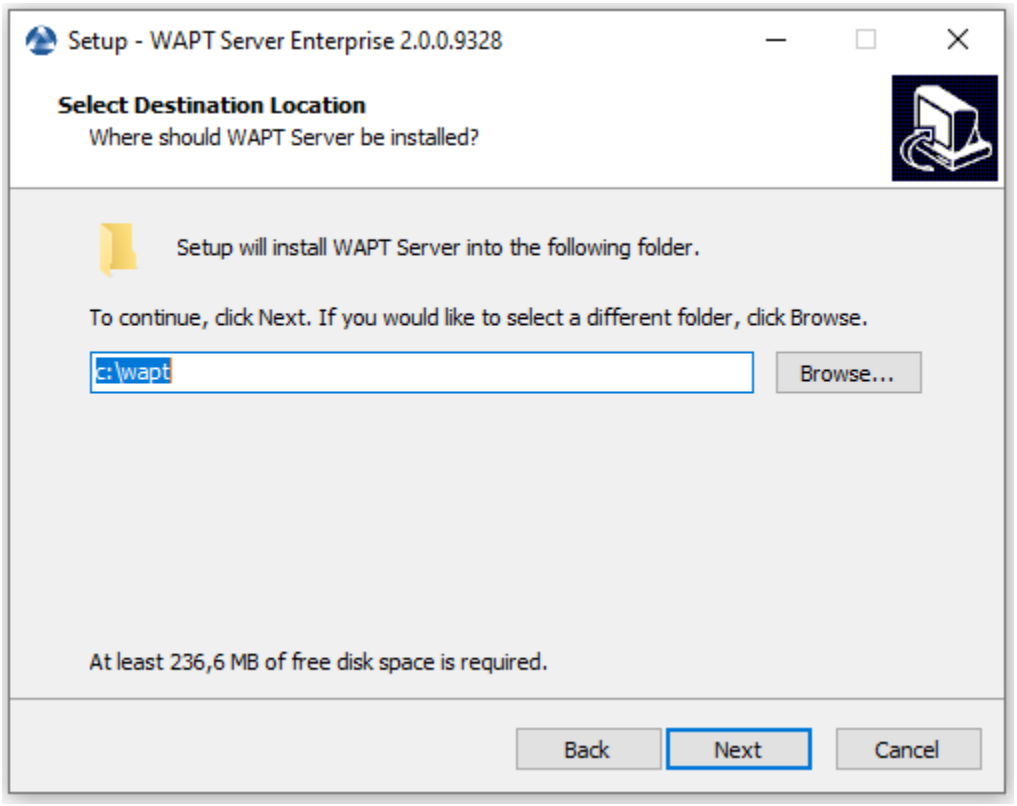


FIG. 3 – Choisissez le dossier de destination WAPT

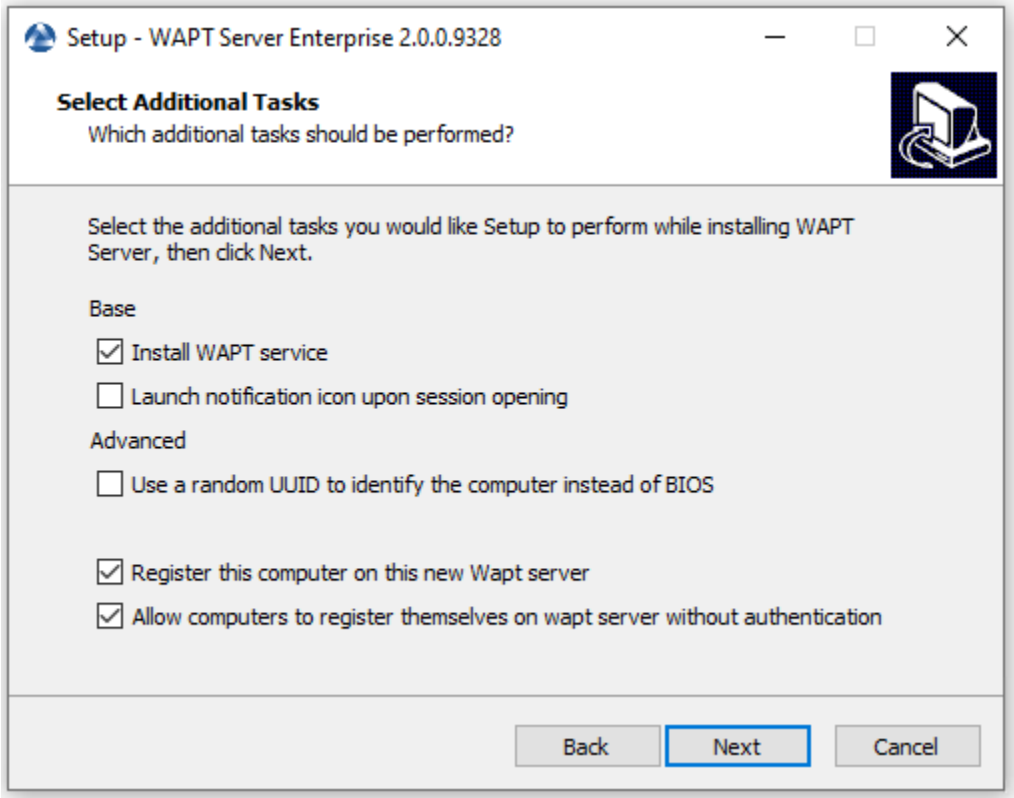


FIG. 4 – Choisissez une tâche supplémentaire

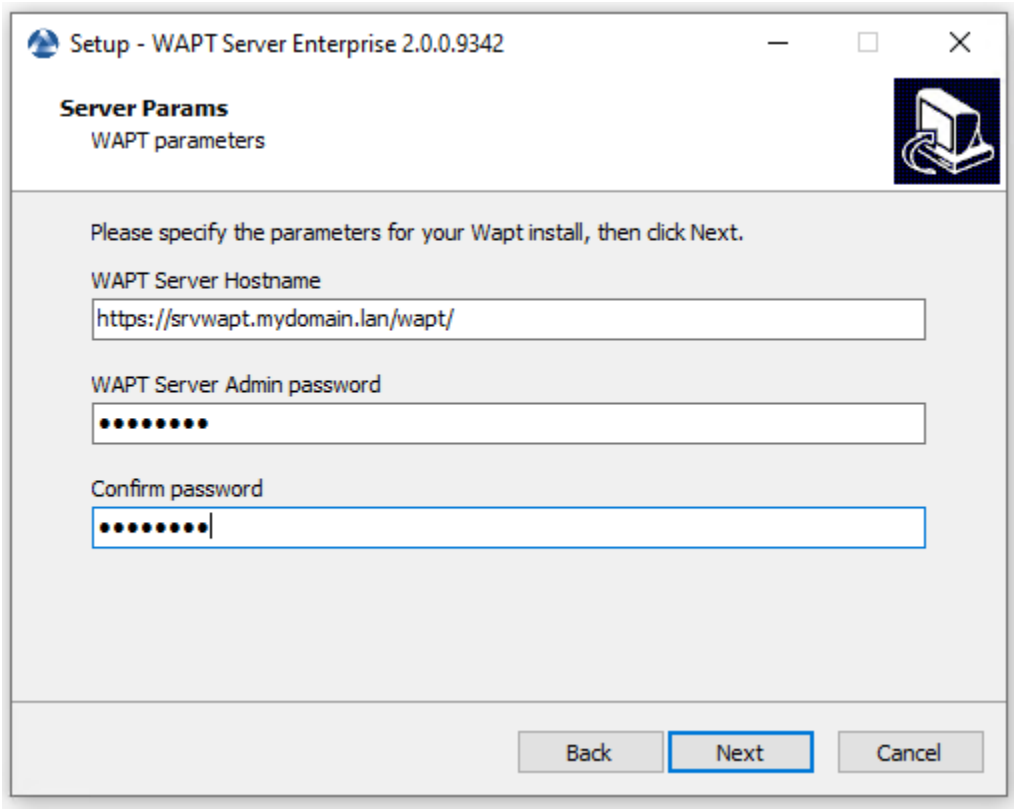


FIG. 5 – Choisir un mot de passe

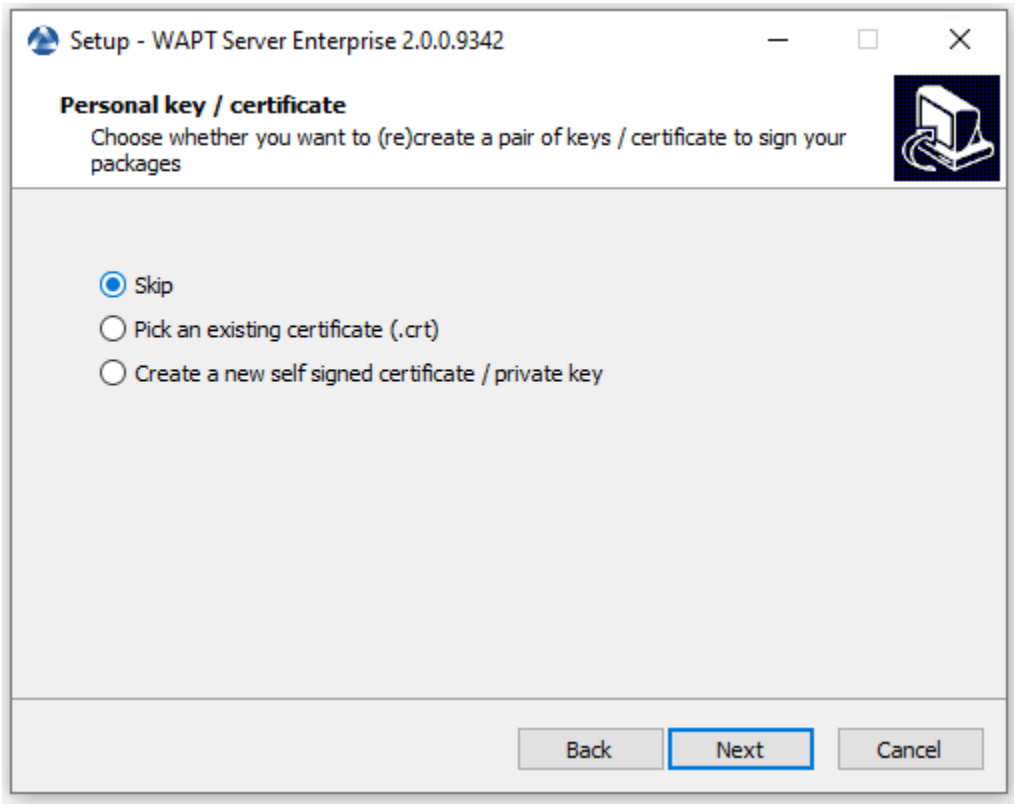


FIG. 6 – Sauter la création de la clé de signature

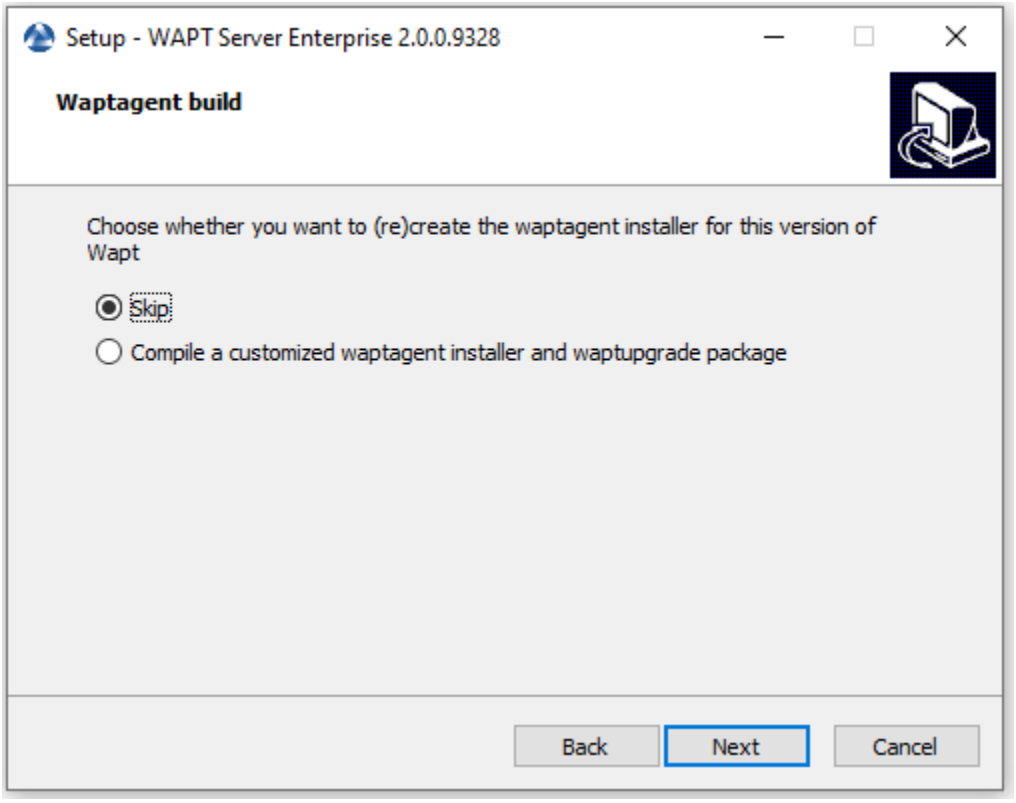


FIG. 7 – Sauter la construction de l'agent WAPT

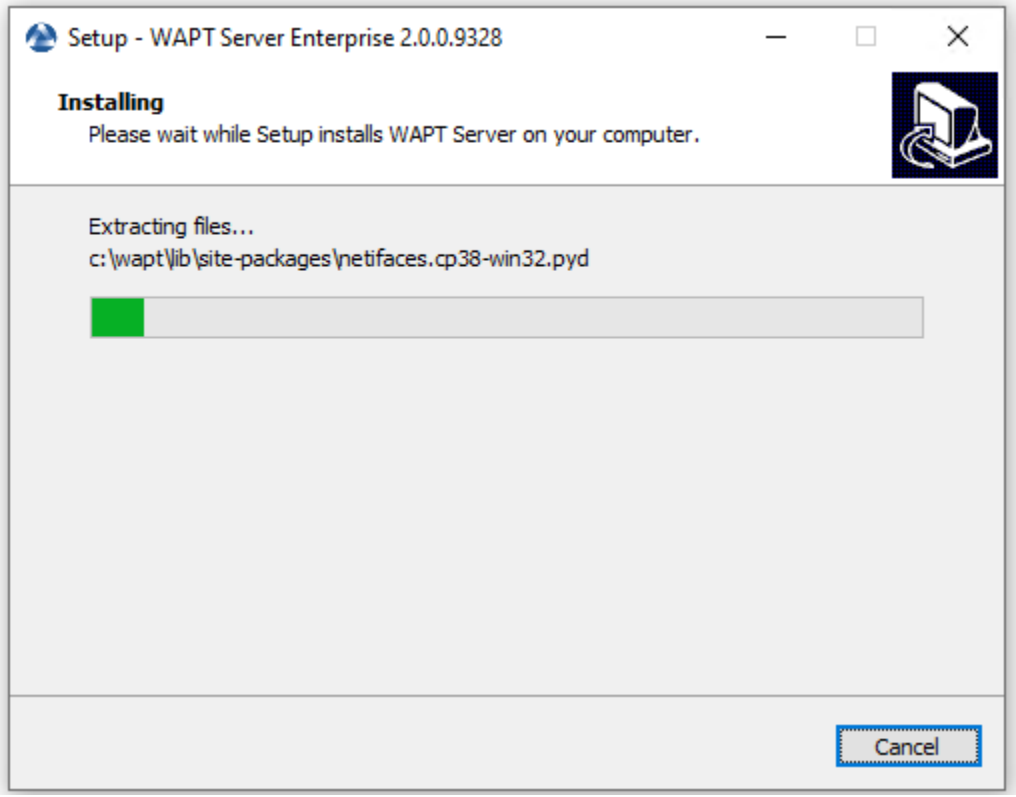


FIG. 8 – Progression de l'installation du serveur WAPT

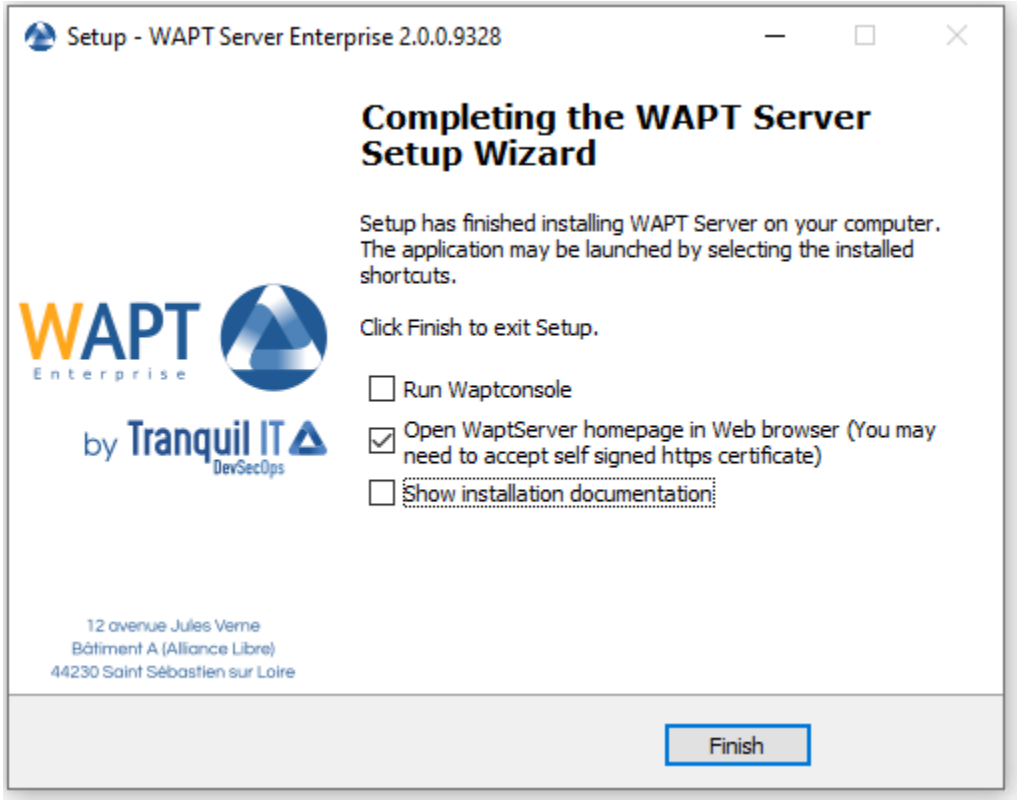


FIG. 9 – L'installation est terminée

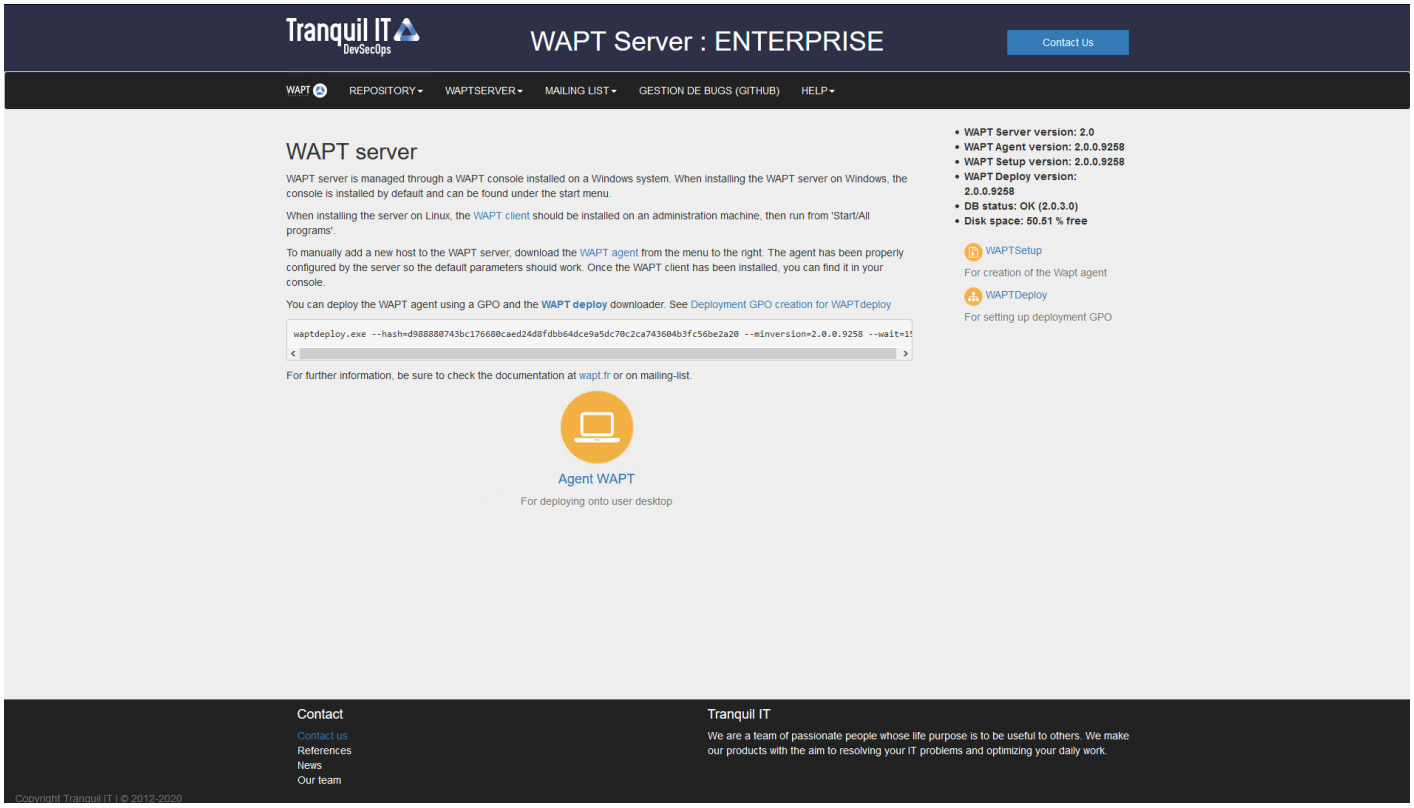


FIG. 10 – Le serveur WAPT sur votre Windows est prêt.

Attention : Pour des raisons de sécurité, n'exécutez pas la console WAPT ou votre outil de développement de paquet WAPT sur le serveur WAPT.

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.

CHAPITRE 13

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.

Avant d'installer la console WAPT, téléchargez-la sur le serveur Tranquil IT :

- Version **Discovery** : WAPT Discovery sera publié ultérieurement. Pour l'instant, l'édition gratuite de WAPT se réfère à la documentation de wapt-1.8 https://www.wapt.fr/en/doc-1.8/
- Version **Enterprise** :
  - Téléchargez `waptsetup.exe` sur 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.

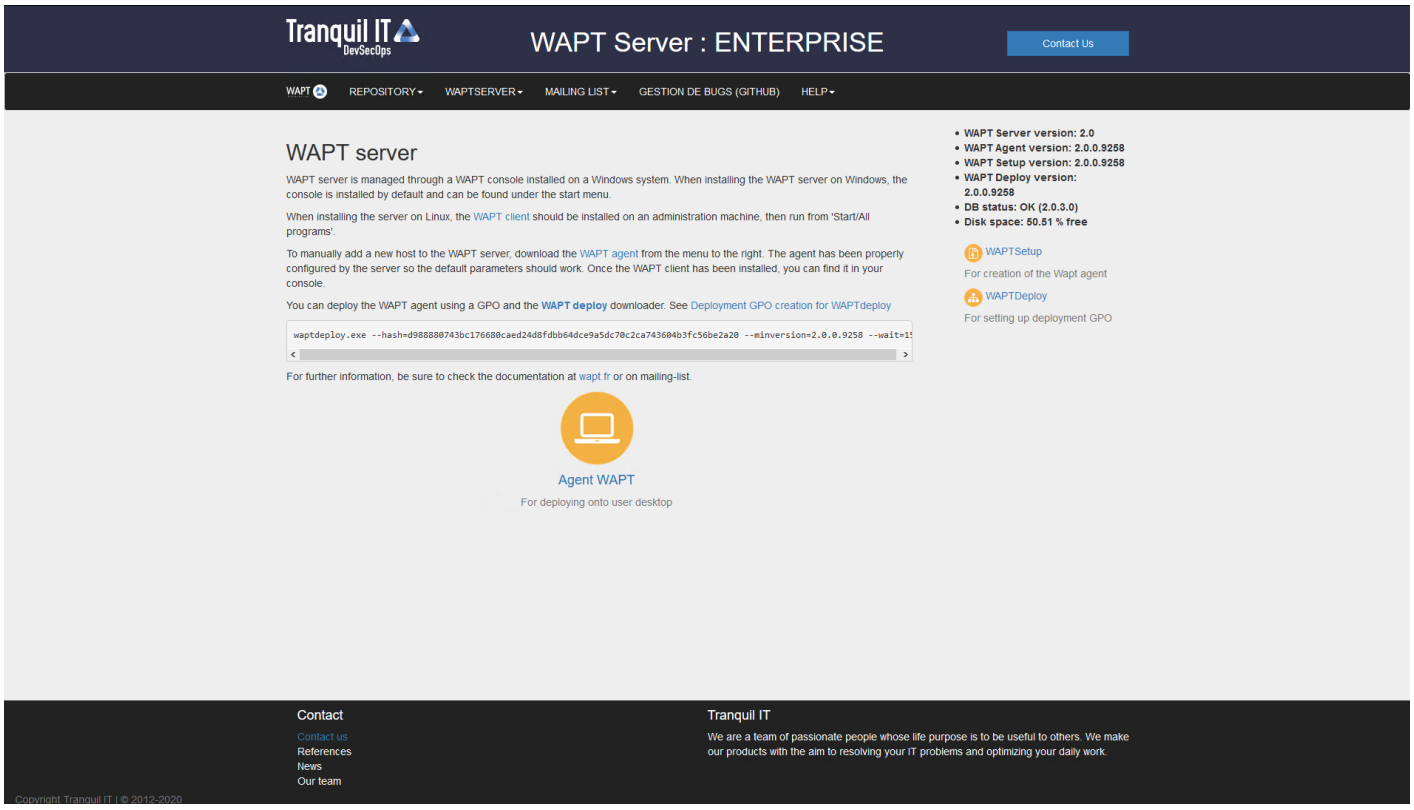


FIG. 1 – L'interface web du serveur WAPT

— 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.

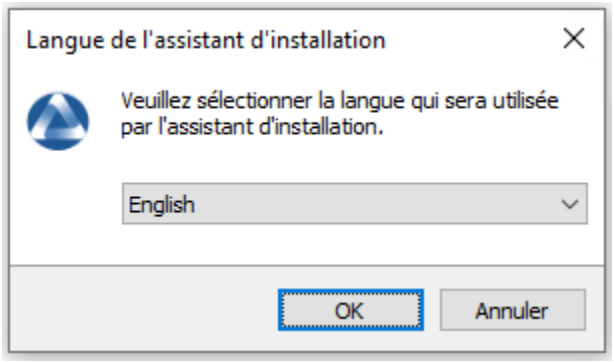


FIG. 2 – Choix de la langue pour WAPT

— 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 installeur	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 *BIOS UUID bugs*	Non coché

— Configurez l'URL du serveur WAPT .

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

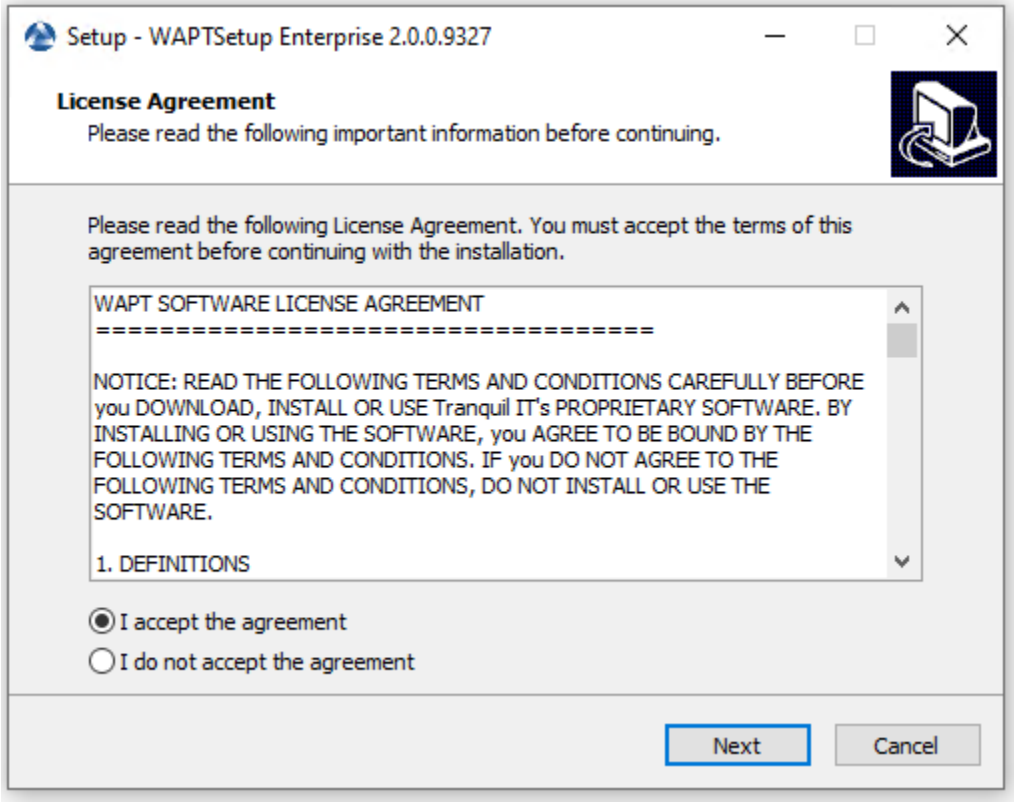


FIG. 3 – Acceptation des conditions de la licence WAPT

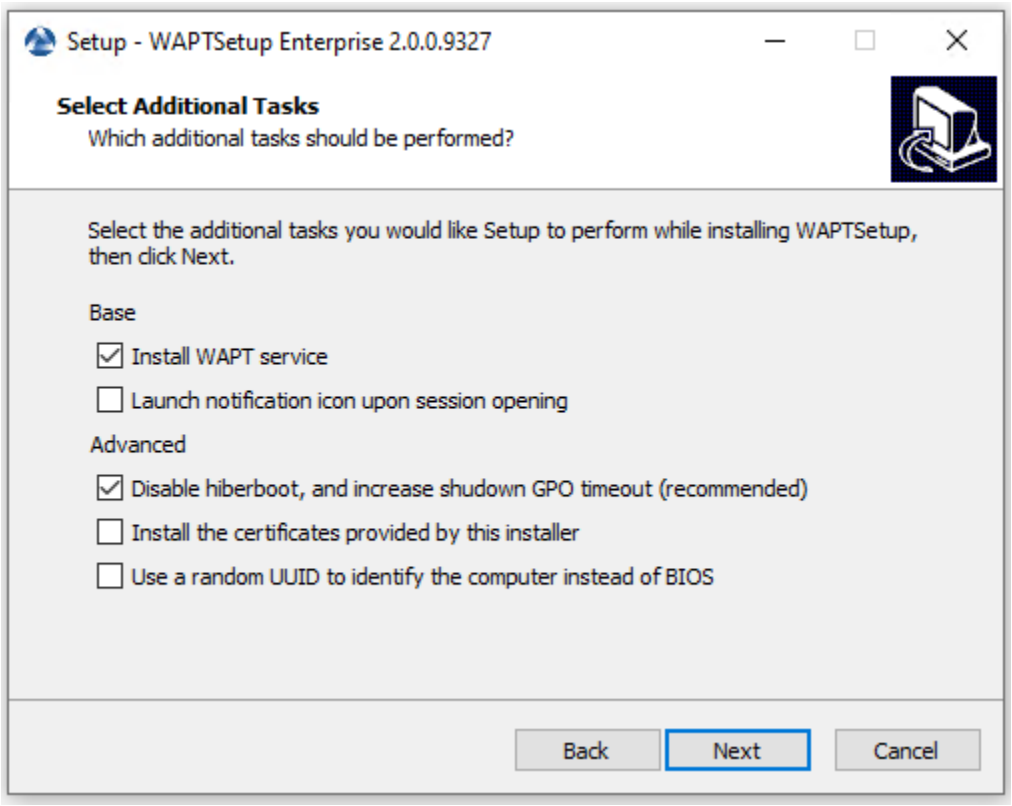


FIG. 4 – Choisir les options de l'installateur

— S'il s'agit de la première installation et que l'agent WAPT.
  — Vérifiez les « Informations statiques WAPT » et définissez-les :
    — URL du dépot WAPT : http://srvwapt.mydomain.lan/wapt.
    — URL du serveur WAPT : https://srvwapt.mydomain.lan.

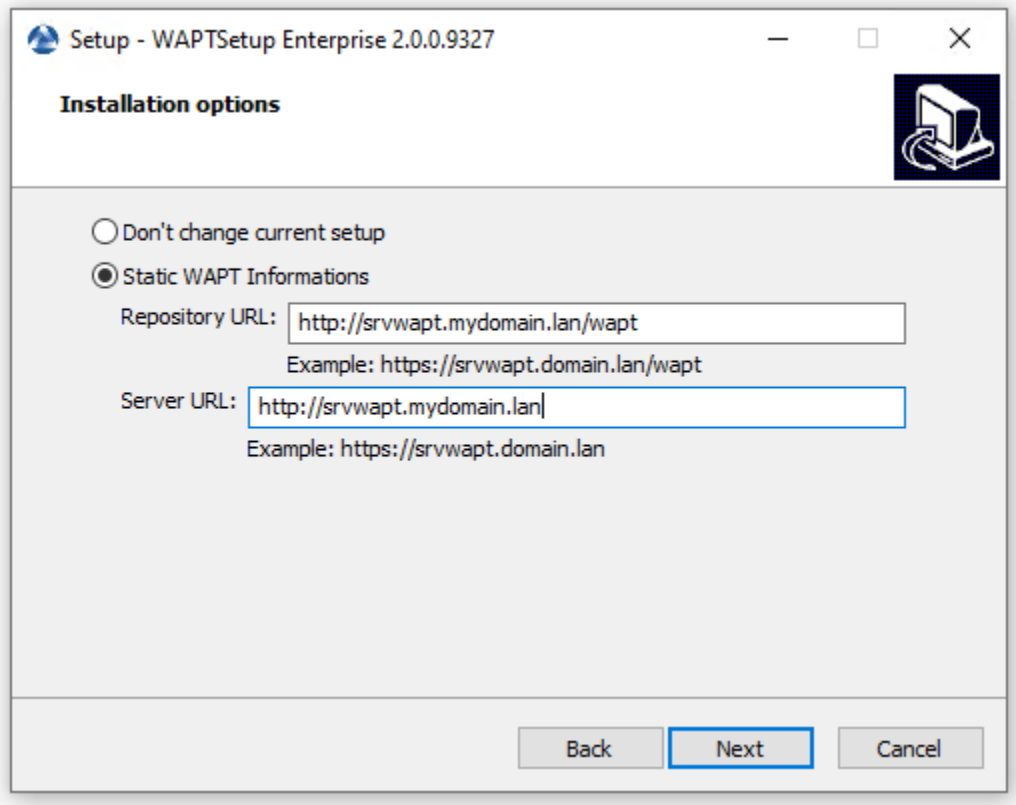


Fig. 5 – Choix du dépot et du serveur WAPT

  — Choisir le dépôt et le serveur WAPT ; cliquer sur *Suivant*.
— Si la console WAPT ou l'agent WAPT est déjà installé :
  — Cochez *Ne pas modifier la configuration actuelle*, puis cliquez sur *Suivant*.
  — 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*.
— Log into the WAPT console with the *SuperAdmin* login and password.

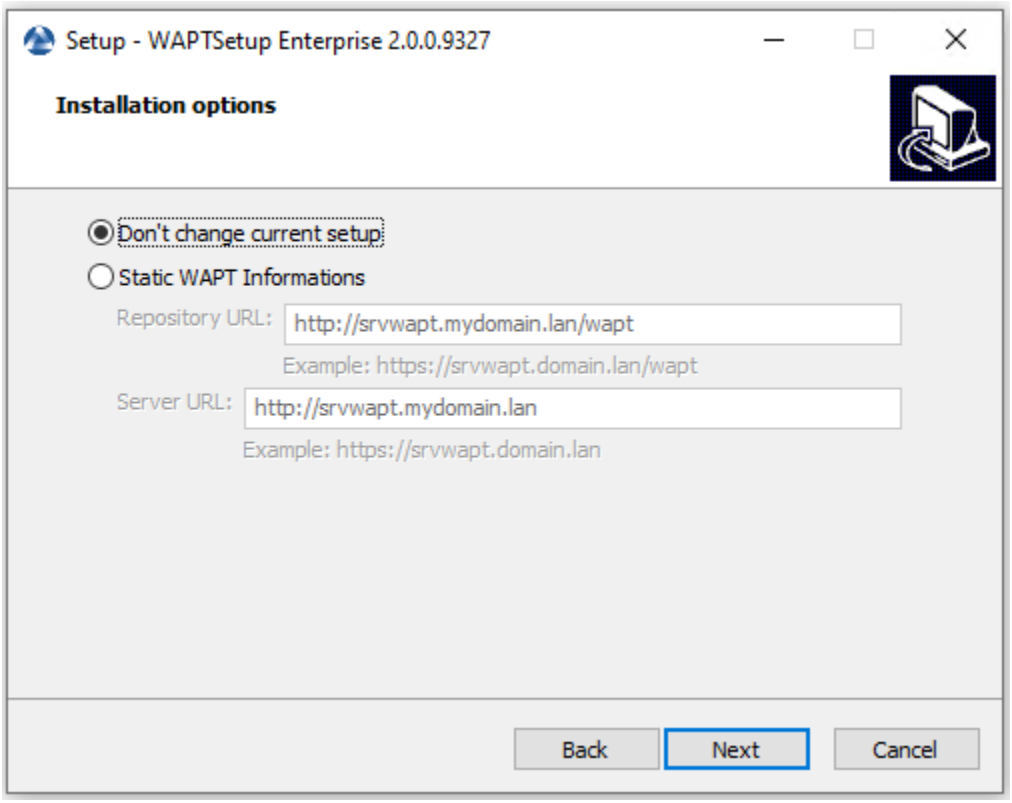
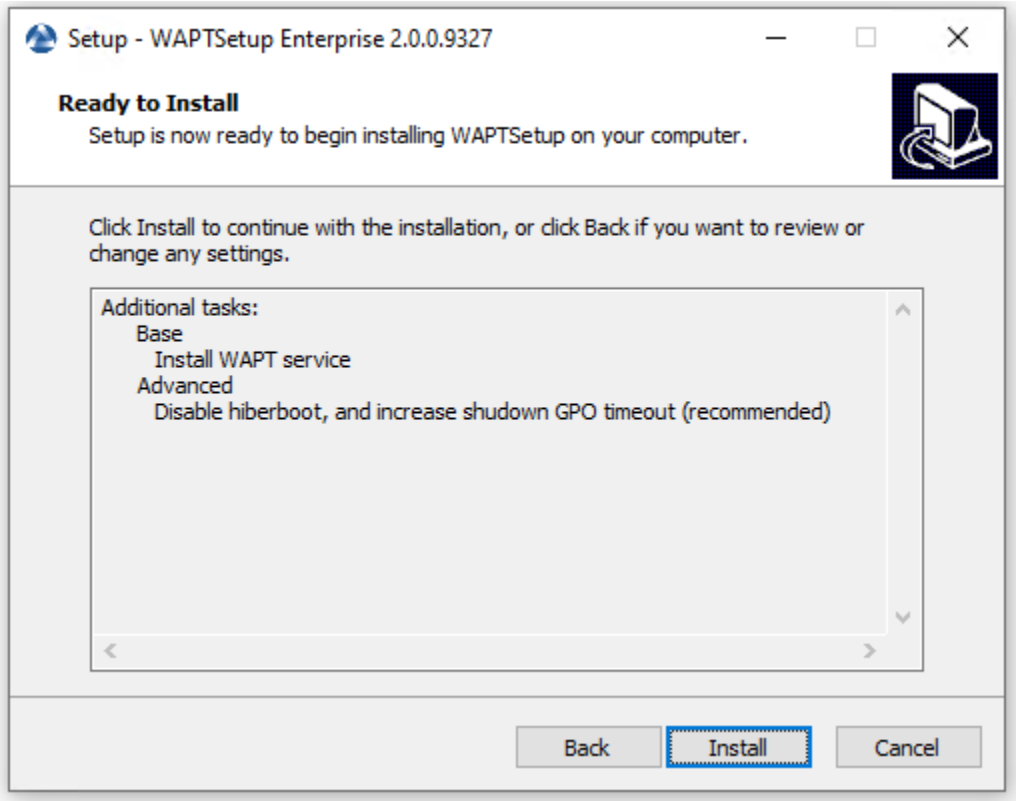


FIG. 6 – Le dépôt et le serveur WAPT sont déjà configurés



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*.

Attention : Dans la version Enterprise, copiez le fichier `licence.lic` que vous avez reçu dans `C:\Program Files (x86)\wapt\licences` pour activer les fonctionnalités Enterprise.

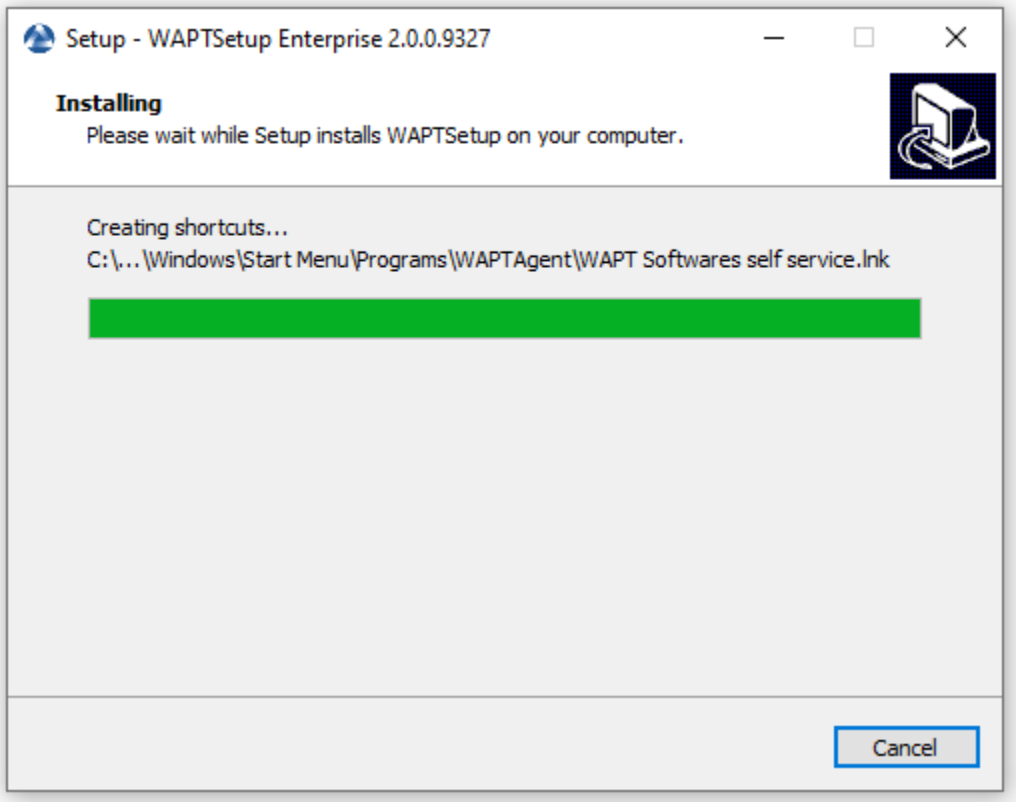


FIG. 7 – Assistant d'installation en cours

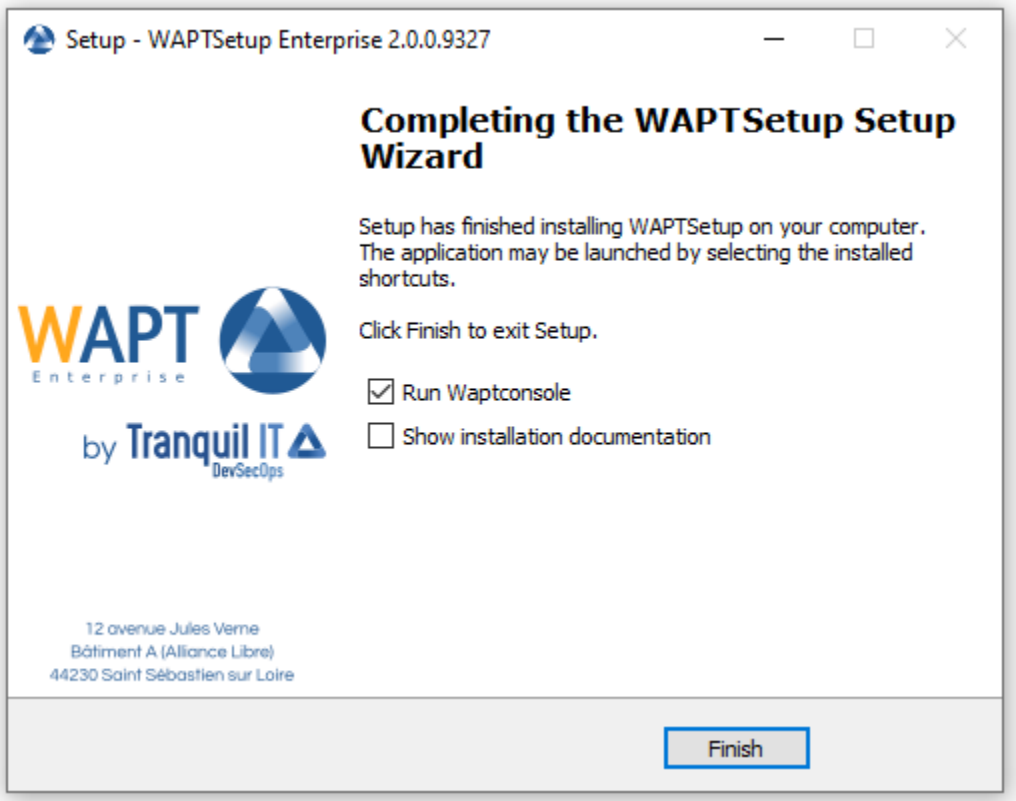


FIG. 8 – L'assistant d'installation est terminé

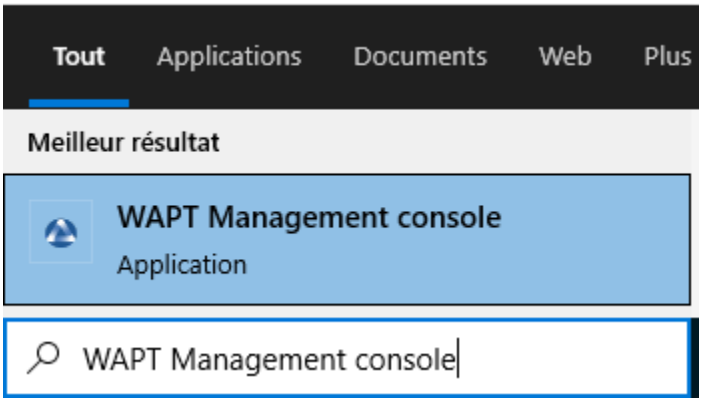


FIG. 9 – Le menu de démarrage de la console WAPT

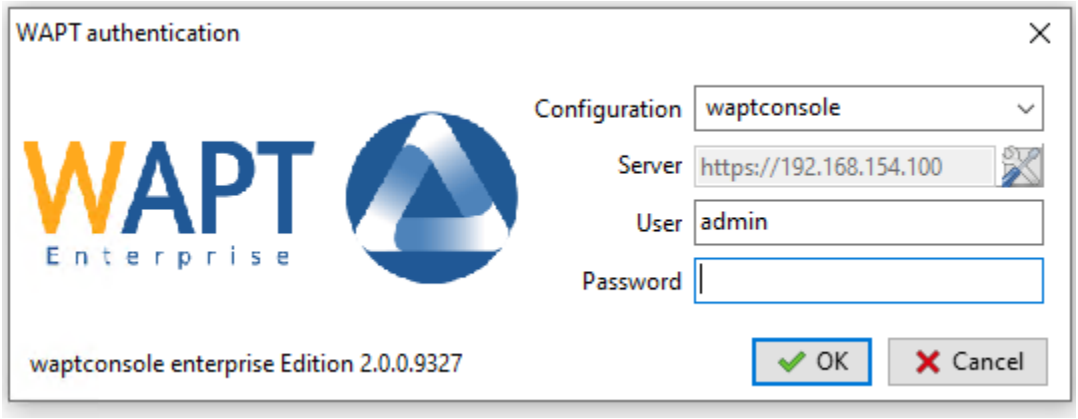


FIG. 10 – The WAPT Console authentication window

13.4.1 Premier démarrage après l'installation du serveur

Indication : On first start, you must start the WAPT console with elevated privileges. *Right-click on the WAPT console binary → Start as Local Administrator*.

Activation de la licence

Attention : Si le message suivant apparaît, vous n'avez pas copié votre `licence.lic` dans `C:\Program Files (x86)\wapt\licences`.

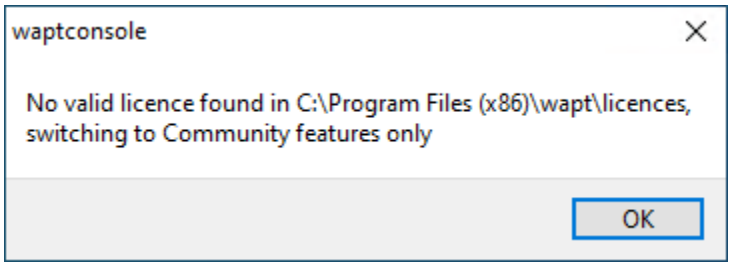


FIG. 11 – WAPT licence not found

Affectation du certificat

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

Voir *l'étape suivante* pour créer votre certificat.

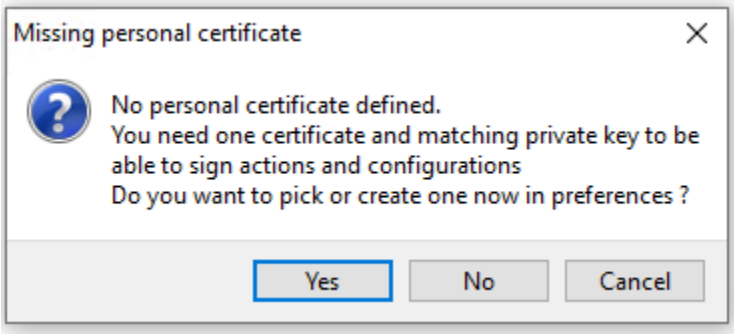


FIG. 12 – WAPT personal certificate not present

Définition du préfixe de paquet

Erreurs de lancement

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

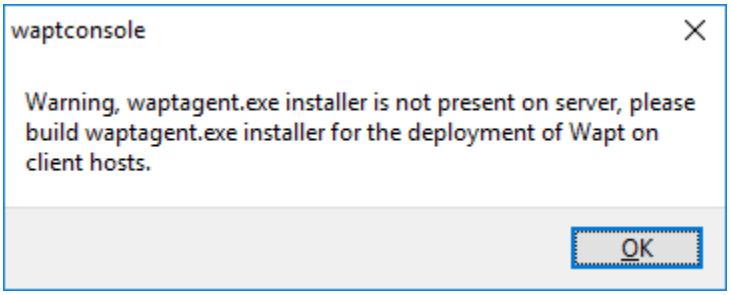


FIG. 13 – WAPT agent not present

CHAPITRE 14

Génération du certificat de l'administrateur pour la signature des paquets WAPT

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`.

14.1 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 dépot WAPT.

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

14.2 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.

14.3 Générer un certificat

In the WAPT console go to *Tools* → *Build certificate*.

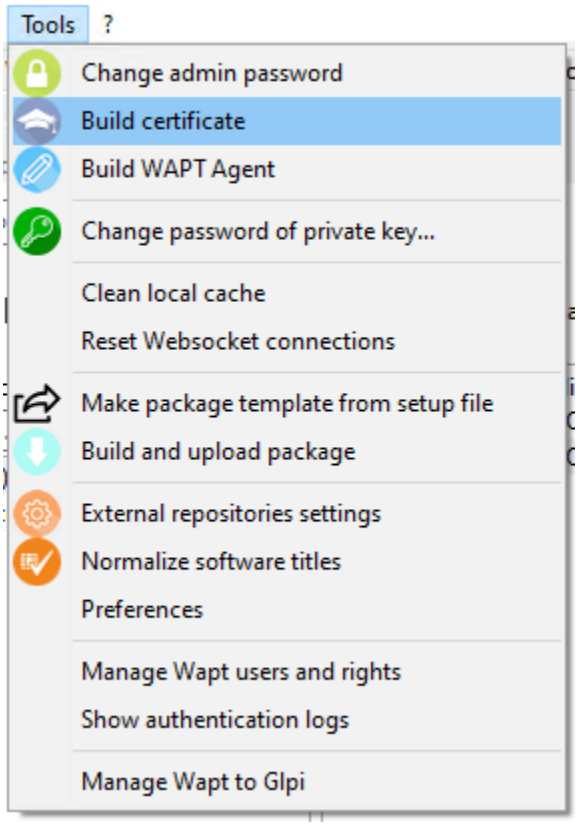


FIG. 1 – Building a self-signed certificate

Important : Nous avons deux options différentes :

— *Créer un certificat pour la version Discovery*.
— *Créer un certificat pour la version Entreprise*.

14.3.1 Discovery

— Remplissez les champs suivants.

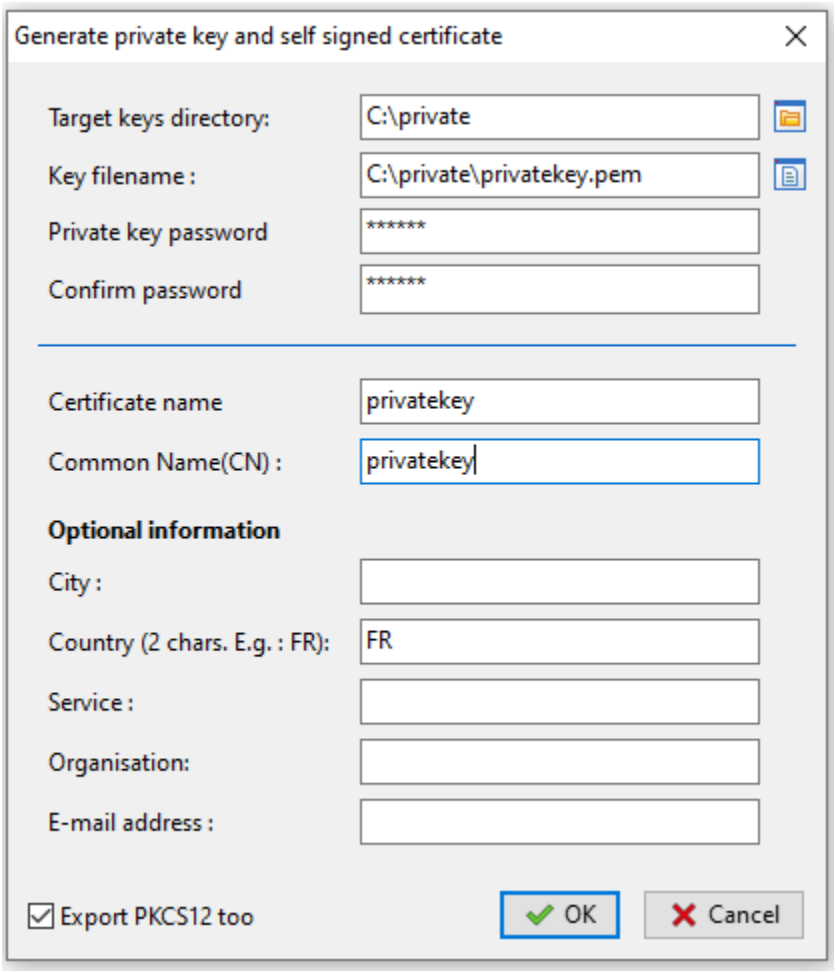


FIG. 2 – Creating a self-signed certificate for the Discovery version

TABLEAU 1 – Informations sur le certificat

Valeur	Description	Requis
Répertoire de destination des clés	Folder where the private key and the public certificate will be stored.	✅
Nom de fichier de la clé	name of the `.pem` and *Name of the private key*.	✅
Mot de passe de la clé	Password for locking and unlocking the key.	✅
Confirmer le mot de passe de la clé	Password confirmation for locking and unlocking the key.	✅
Nom du certificat	Name of the `.crt` certificate.	✅
Common Name (CN)	Display name of the certificate.	✅
Ville	Name of the certificate holder's city to register in the certificate.	❌
Pays (2 caractères. Exemple : FR)	Name of the certificate holder's country (FR, EN, ES, DE ...) to register in the certificate.	❌
Service	Name of certificate holder's service or organizational department to register in the certificate.	❌
Organisation	Name of the certificate holder's Organization to register in the certificate.	❌
Adresse E-Mail	Email address of the certificate holder to register in the certificate.	❌
Export PKCS12	Create `*.p12` certificate in *Target key directory*.	❌ (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 *Administrator* 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 :

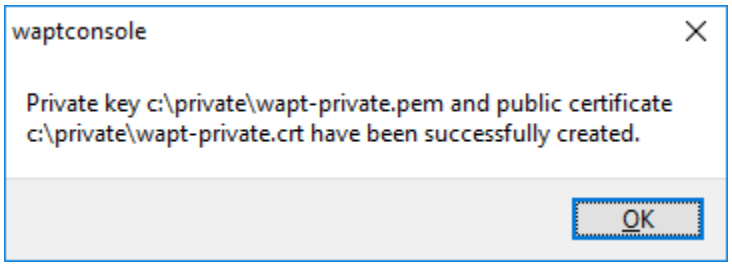


FIG. 3 – Certificate generated successfully

— Cliquez sur *OK*.

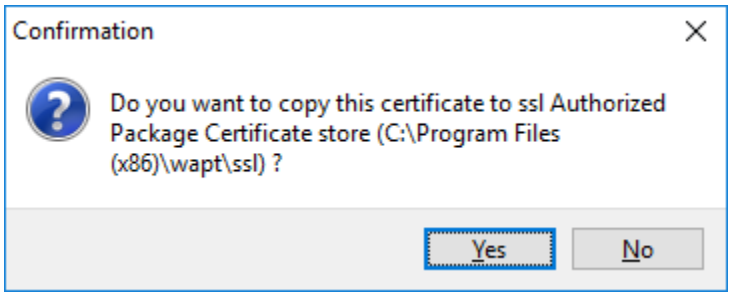


FIG. 4 – Confirmation de la copie du certificat dans le dossier ssl

— 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*.

14.3.2 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*.

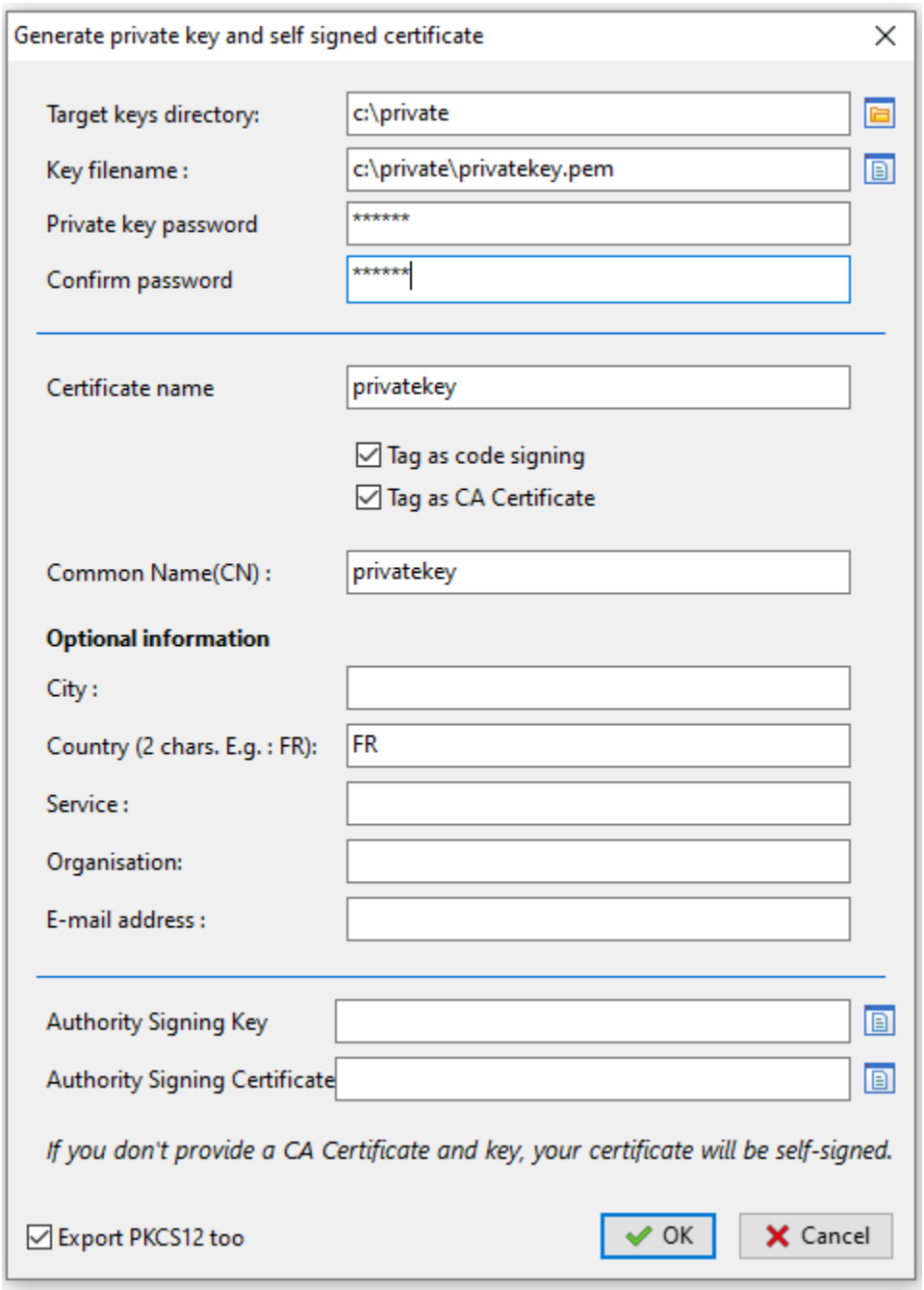


FIG. 5 – Creating a self-signed certificate for Enterprise version

Tableau 2 – Informations sur le certificat

Valeur	Description	Requis
Répertoire de destination des clés	Folder where the private key and the public certificate will be stored.	✅
Nom de fichier de la clé	name of the `.pem` and *Name of the private key*.	✅
Mot de passe de la clé	Password for locking and unlocking the key.	✅
Confirmer le mot de passe de la clé	Password confirmation for locking and unlocking the key.	✅
:guilabel :`Pour Signature de code`	Check this box if the certificate/ key pair will be allowed to sign software packages.	✅
Pour usage en tant que CA	Check this box if this certificate can be used to sign other certificates (main or intermediate Certificate Authority).	✅
Nom du certificat	Name of the `.crt` certificate.	✅
Common Name (CN)	Display name of the certificate.	✅
Ville	Name of the certificate holder's city to register in the certificate.	❌
Pays (2 caractères. Exemple : FR)	Name of the certificate holder's country (FR, EN, ES, DE ...) to register in the certificate.	❌
Service	Name of certificate holder's service or organizational department to register in the certificate.	❌
Organisation	Name of the certificate holder's Organization to register in the certificate.	❌
Adresse E-Mail	Email address of the certificate holder to register in the certificate	❌
Clé privée de l'autorité	Clé privée (`*.pem`) de l'autorité de certification (CA)	❌
Certificat de l'autorité	Certificat (`*.crt`) de l'autorithé de certificat (CA)	❌
Export PKCS12	Créer un certificat `*.p12` dans *répertoire de destination des clés*	❌ (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**.

For creating a Certificate Authority, go to the section on *generating the Certificate Authority (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 :

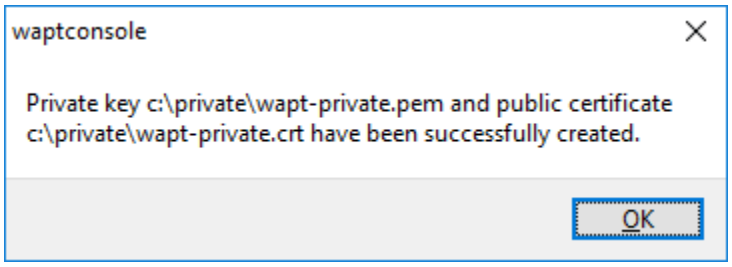


FIG. 6 – Certificate generated successfully

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

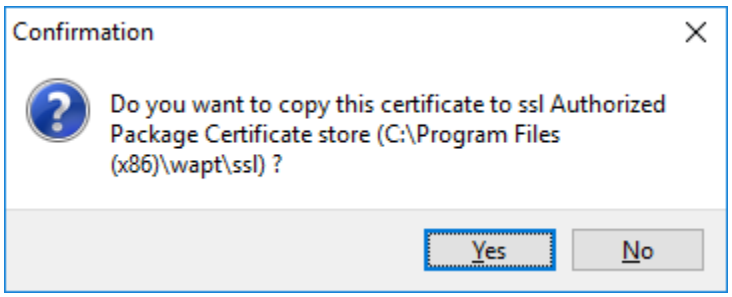


FIG. 7 – Confirmation de la copie du certificat dans le dossier ssl

— 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*.

CHAPITRE 15

Construction du 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.

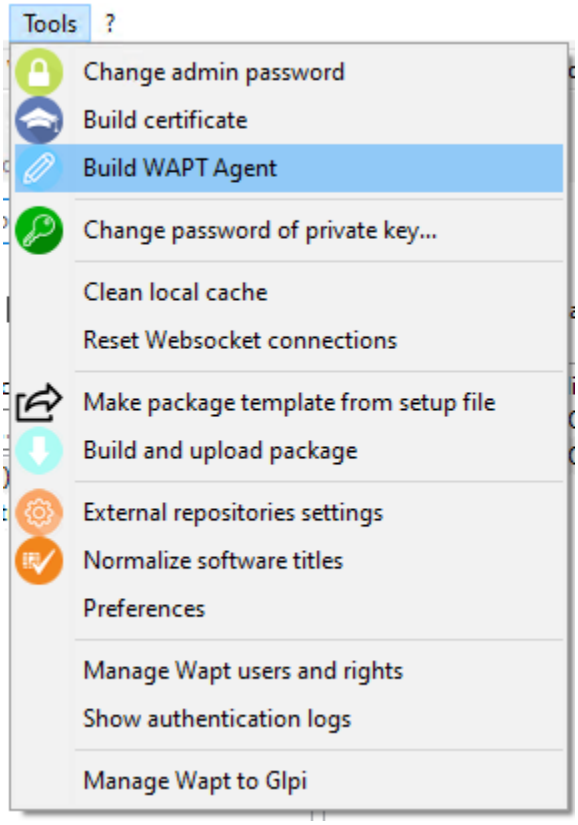


FIG. 1 – Generating the WAPT agent from the console

15.1 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 !

15.1.1 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 ! ! !

15.1.2 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 !

15.1.3 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.

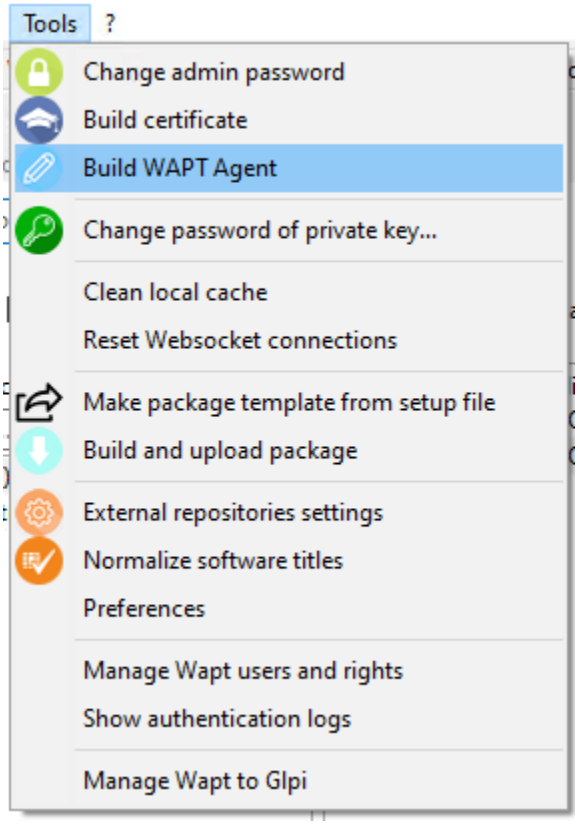


FIG. 2 – Generating the WAPT agent from the console

15.2 Discovery

Tableau 1 – Informations sur l'agent WAPT

Valeur	Description	Requis
Liste de certificats autorisés	Dossier du certificat de confiance.	✓
Inclure également les certificats utilisateurs	Inclure le certificat WAPT local.	✗
Adresse principale du dépôt WAPT	Adresse du dépôt sur le serveur WAPT.	✓
Adresse du serveur WAPT	Adresse du dépôt sur le serveur WAPT.	✓
Chemin vers les certificats d'autorité pour les serveurs https	Chemin d'accès aux certificats utilisés pour la vérification HTTPS.	✗
Organisation	Name of the Organization to identify the origin of WAPT packages.	✗
Utiliser le nom FQDN de la machine comme identifiant UUID	Si des FQDN sont utilisés pour *identifier les agents WAPT*.	✗
Utiliser un UUID de machine aléatoire (pour les BIOS buggés)	Si des UUID aléatoires sont utilisés pour *identifier les agents 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 **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.

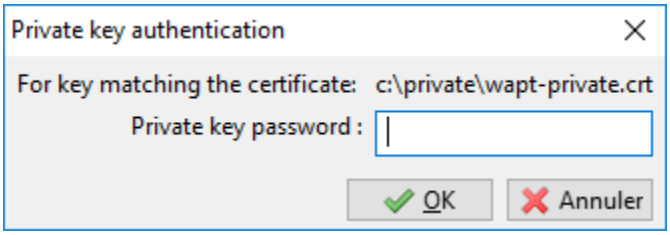


Fig. 3 – Entering the password for unlocking the private key

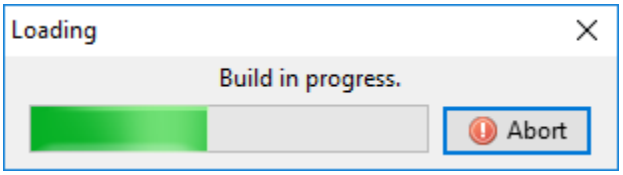


Fig. 4 – Progression de la construction de l'agent installateur WAPT

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

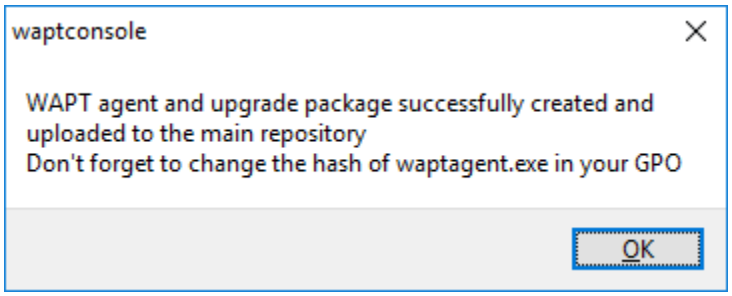


FIG. 5 – Confirmation of the WAPT agent loading onto WAPT repository

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.

15.3 Enterprise

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

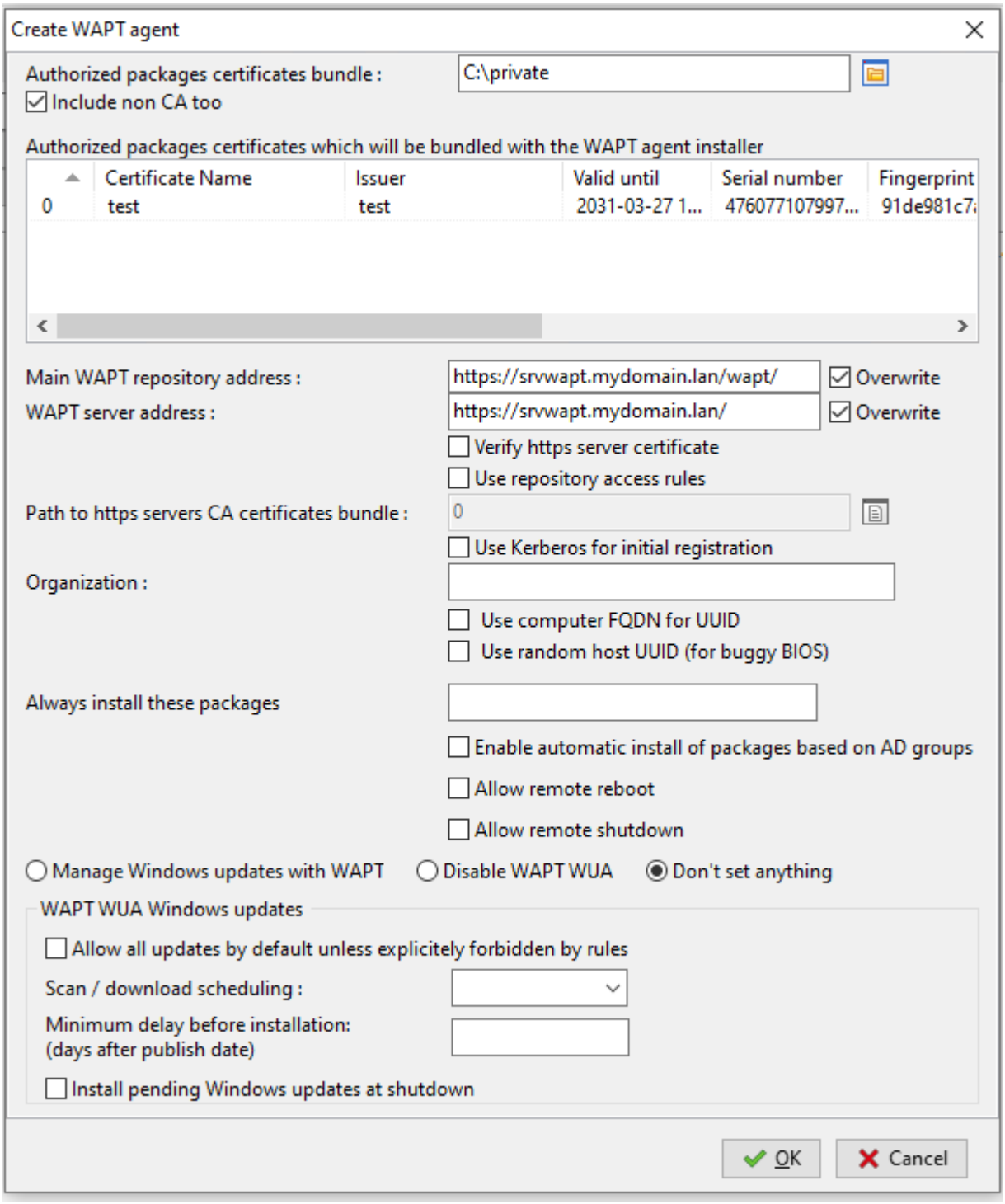


FIG. 6 – Filling in the informations on your Organization

TABLEAU 2 – Informations sur l'agent WAPT

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

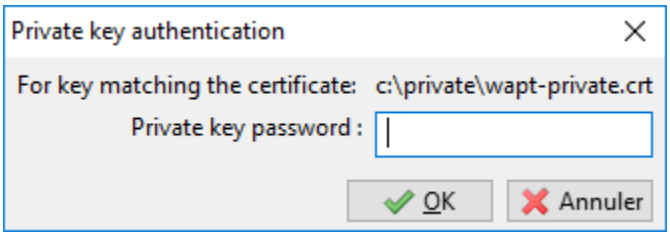


Fig. 7 – Providing the password for unlocking the private key

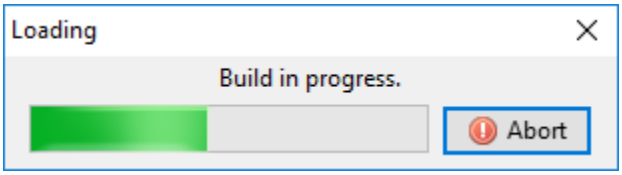


Fig. 8 – Progression de la construction de l'agent installateur WAPT

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

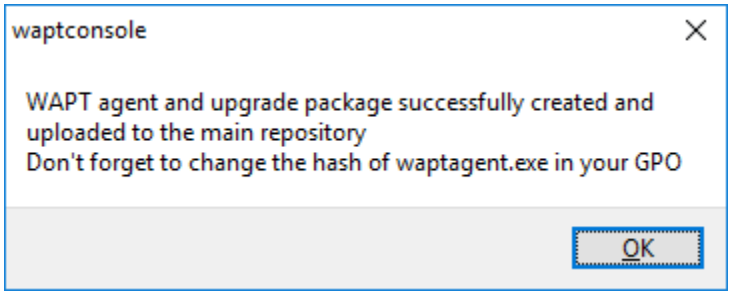


Fig. 9 – Confirmation of the WAPT agent loading onto WAPT repository

---

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.

CHAPITRE 16

Configuration de l'agent WAPT avec des options avancées

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)

(suite de la page précédente)

```
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)
```

16.1 Description des sections disponibles

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

Section	Description
[global]	Global WAPT agent options.
[wapt]	Main repository options.
[wapt-template]	External remote repository options.
[wapt-host]	Repository for host packages options.
[waptwua]	WUA agent options.
[repo-sync]	For synching multiple repositories.

Toutes les sections sont détaillées ci-dessous.

16.2 Description des options disponibles par section

16.2.1 [global]

General settings

Tableau 3 – Description of available options for the WAPT agent in the [global] section

Options / Valeur par défaut	Description	Exemple
allow_remote_reboot = False	Allows *rebooting hosts remotely* from the WAPT console (default False).	allow_remote_reboot = True
allow_remote_shutdown = False	Allows *shutting down the host remotely* from the WAPT console (default False).	allow_remote_reboot = True
check_certificates_validity = False	Enable verification 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_with_task_period = True	Définit si un téléchargement de paquets en attente doit être lancé après une mise à jour avec waptupdate_task_period.	download_after_update_with_ = False
host_organizational_unit_dn = None	Permet de forcer une unité organisationnelle sur l'agent WAPT (pratique pour attribuer un *faked OU* 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_dn = OU=TOTO,OU=TEST,DC=MYDO
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	Allows to set the list of WAPT agent languages to pre-filter the list of packages visible by the WAPT agent (for package filtering). The parameter accepts multiple entries ordered by preference (eg. locales=fr,en).	locales = en
log_to_windows_events = False	Envoie 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 *ici*.	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	Sets how the self service authentication works. Possible values are : system, waptserver-ldap or waptagent-ldap	service_auth_type = waptserver-ldap
uninstall_allowed = True	Defines whether or not it is possible for the user to uninstall applications via the self-service.	uninstall_allowed = False
use_ad_groups = True	For using *group packages* (default False).	use_ad_groups = True
use_fqdn_as_uuid	Allows to use the FQDN rather than the BIOS UUID as the unique machine identi-	use_fqdn_as_uuid

Note :
— If there is no `repo_url` attribute in the [global] section, then a repository in the `[wapt]` section will have to be explicitly defined. It will have to be enabled by adding it to the `repositories` attribute.
— If there is no `wapt_server` attribute in the [global] section, then no WAPT Server will be used.

Paramètres du serveur

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

Tableau 4 – Description of available options for the WAPT agent in the [global] section for server configuration

Options / Valeur par défaut	Description	Exemple
`public_certs_dir` = None	Dossier des certificats autorisés à vérifier la signature des paquets WAPT.	`public_certs_dir = C:\Program Files (x86)\wapt\ssl` (sous Windows) `public_certs_dir = /opt/wapt/ssl/` (sous Linux et MacOS)
`use_kerberos` = False	Use kerberos authentication for initial registration on the WAPT Server (default False).	`use_kerberos` = True
`verify_cert` = False	See the documentation on activating the *verification of HTTPS certificates*	`verify_cert` = True
`wapt_server` = None	URL du serveur WAPT. Si cet attribut n'est pas présent, aucun serveur WAPT ne sera contacté.	`wapt_server` = https://srvwapt.mydomain.lan
`wapt_server_timeout` = 30	Délai de connexion HTTPS du serveur WAPT en secondes	`wapt_server_timeout` = 10

paramètres waptexit

Tableau 5 – Description of available options for the WAPT agent in the [global] section for waptexit

Options / Valeur par défaut	Description	Exemple
`allow_cancel_upgrade` = True (default True)	Prevents users from canceling package upgrades on computer shutdown. If disabled, users will not be able to cancel an upgrade on computer shutdown.	`allow_cancel_upgrade` = True
`hiberboot_enabled` = True (default None)	Désactive Hiberboot sur Windows 10 pour que **`waptexit`** fonctionne correctement.	`hiberboot_enabled` = True
`max_gpo_script_wait` = None (default None)	Délai d'exécution des GPO à l'arrêt de l'ordinateur.	`max_gpo_script_wait` = 180 (default None)
`pre_shutdown_timeout` = None	Délai d'attente pour les scripts à l'arrêt de l'ordinateur.	`pre_shutdown_timeout` = 180

Paramètres d'authentification de Self-Service WAPT et Waptservice

Tableau 6 – Description of available options for the WAPT agent in the [global] section for the WAPT Self-Service and Waptservice Authentification

Options / Valeur par défaut	Description	Exemple
`ldap_auth_base_dn` = None	Utile avec *waptagent-ldap*, définit le *dn de base* pour la requête LDAP.	`ldap_auth_base_dn` = dc=domain,dc=lan
`ldap_auth_ssl_enabled` = False	Utile avec *waptagent-ldap*, définit si la requête LDAP doit être chiffré.	`ldap_auth_ssl_enabled` = True
`ldap_auth_server` = None	Utile avec *waptagent-ldap*, 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 *system*, *waptserver-ldap*, *waptagent-ldap*.	`service_auth_type` = waptagent-ldap
`verify_cert_ldap` = False	Utile avec *waptagent-ldap*, définit si le certificat doit être vérifié.	`verify_cert_ldap` = True
`waptservice_admin_filter` = False	Appliquer un filtrage d'affichage pour les *paquets self-service* pour les administrateurs locaux.	`waptservice_admin_filter` = True
`waptservice_password` = None	mot de passe haché en sha256 lorsque *waptservice_user* est utilisé (la valeur *NOPASSWORD* 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 of available options for the WAPT agent in the [global] section for the wapttray

Options / Valeur par défaut	Description	Exemple
`notify_user` = False	Empêche `wapttray` d'envoyer des notifications (popup).	`notify_user` = True

Paramètres du proxy

Tableau 8 – Description of available options for the WAPT agent in the [global] section for the proxy

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

Paramètres de création des paquets

Tableau 9 – Description of available options for the WAPT agent in the [global] section for creating WAPT packages

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

16.2.2 [wapt-wua]

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

16.2.3 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`.

For information on configuring the WAPT console, please refer to *this documentation*.

[wapt]

Indication : If this section does not exist, parameters are read from the `[global]` section.

[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*.

CHAPITRE 17

Configuration de la console WAPT

Indication : the WAPT console configuration is stored in 2 locations :

— `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* ;

17.1 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 *dépôt Tranquil IT* 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.

17.2 Description des options disponibles par section

17.2.1 [global]

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

Tableau 2: Description of available options in `AppData\Local`

Options / Valeur par défaut	Description	Exemple
`advanced_mode` = False	Lance la console en mode débogage.	
`allow_remote_reboot` = False	Allows *rebooting hosts remotely* from the WAPT console (de	
`allow_remote_shutdown` = False	Allows *shutting down the host remotely* from the WAPT cons	
`client_certificate` = None	Si le dépôt distant utilise l'authentification SSL côté client	
`client_private_key` = None	Si le dépôt distant utilise l'authentification SSL côté client	
`check_certificates_validity` = False	Force la vérification de la date et de la CRL du certificat du p	
`default_maturity` = ""	Maturité de téléchargement par défaut pour les paquets WAP	
`default_package_prefix` = tis	Préfixe utilisé pour nommer les paquets WAPT.	
`default_sources_root` = `C:\waptdev` (Windows) ou `~/waptdev` (Linux)	Dossier de développement du paquet de base WAPT.	
`grid_hosts_plugins` = W10=	*Plugins externes* pour la console WAPT. La valeur par défaut	
`host_profiles` = None	Permet de définir une liste de paquets WAPT que l'agent WA	
`hiberboot_enabled` = False	Désactive Hiberboot sur Windows 10 pour **`waptexit`**	
`http_proxy` = None	Adresse du serveur proxy dans la console WAPT.	
`last_usage_report` = ""	Date de la dernière utilisation de la console WAPT.	
`lastwaptserveruser` = ""	Dernier utilisateur connecté sur cette console WAPT.	
`max_gpo_script_wait` = 180	Délai d'exécution des GPO à l'arrêt de l'ordinateur.	
`personal_certificate_path` = ""	Chemin d'accès au certificat associé à la clé privée de l'admi	
`pre_shutdown_timeout` = 180	Délai d'attente pour les scripts à l'arrêt de l'ordinateur.	
`repo_url` = l'adresse de votre dépôt WAPT	Adresse du dépôt principal de WAPT.	
`send_usage_report` = True	Allows the WAPT console to send anonymous statistics to Tr	
`sign_digests` = sha256	Liste des algorithmes de signature autorisés pour les paquets	
`use_ad_groups` = False	Pour utiliser les *paquets unit*.	
`use_fqdn_as_uuid` = False	Permet d'utiliser le FQDN plutôt que l'UUID du BIOS comm	
`use_kerberos` = False	Utilisez l'authentification kerberos pour l'enregistrement init	
`use_hostpackages` = False	Utilisation des *paquets host*.	
`use_http_proxy_for_repo` = False	Utilisez un proxy pour vous connecter au dépôt principal de	
`use_http_proxy_for_server` = False	Utilisez un proxy pour vous connecter au serveur WAPT à pa	
`use_repo_rules` = False	Pour les *dépôts secondaires*.	
`verify_cert` = False	For *verifying SSL / TLS certificate*.	
`wapt_server` = ""	Adresse du serveur WAPT.	

Tableau 3 – Description of available options on 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 autosignés 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 *Données d'audit* sur l'inventaire des machines.	show_host_audit_data_tab = True
use_ad_groups = False	For using *unit packages* (default False).	use_ad_groups = True
use_fqdn_as_uuid = False	Allows you to use the fqdn name rather than the uuid BIOS as the unique machine identifier in wapt (default False).	use_fqdn_as_uuid = True
waptconsole.version = ""	Version de la console	waptconsole.version = 2.0.0.9424
waptwua_enabled = False	For displaying Windows Update tab on console	waptwua_enabled = True

17.2.2 [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*.

See available parameters and configurations by visiting *this documentation on setting up multiple repositories*.

CHAPITRE 18

Configuration du serveur WAPT

The WAPT Server configuration file on GNU/ Linux and macOS systems is found in `/opt/wapt/conf/waptserver.ini` or in `/opt/wapt/waptserver/waptserver.ini`.

The WAPT Server configuration file on Windows is found in `C:\wapt\conf\waptserver.ini`.

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

18.1 Section [option] of waptserver.ini

Several options can be defined in the [option] section.

```
[options]
```

Tableau 1: Available parameters for the [option] section of `waptserver.ini`

Options / Valeur par défaut	Description	Exemple
`allow_unauthenticated_connect` = None	Définit si les connex	
`allow_unauthenticated_registration` = False	Permet l'enregistrer	
`allow_unsigned_status_data` = False	Débogage uniquem	
`application_root` = ""	Définit un chemin r	
`auto_create_ldap_users` = True	Relatif aux ACL uti	
`client_certificate_lifetime` = 3650	Définit la durée de v	
`clients_read_timeout` = 5	Définit le délai d'at	
`clients_signing_certificate` = None	Définit le chemin du	
`clients_signing_crl_days` = 30	Définit la durée de v	

suite sur la page suivante

Tableau 1 – suite de la page précédente

Options / Valeur par défaut	Description	Exemple
clients_signing_crl = None	Définit le chemin d	
clients_signing_crl_url = None	Définit l'URL de la	
clients_signing_key = None	Définit le chemin d	
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 ma	
htpasswd_path = None	Ajoute l'authentific	
http_proxy = None	Définit le serveur pr	
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 **HTT**	
nginx_https = 443	Définit le port **HTT**	
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 W	
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 du	
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	

18.2 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)

(suite de la page précédente)

```
  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;
    }
  }
}
```

18.3 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.

18.3.1 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
```

18.3.2 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
```

18.3.3 Configuration de la base de données PostgreSQL

A higher number of clients need a higher number of connections to the PostgreSQL database. In the `postgresql.conf` file (file *:/etc/postgresql/{version}/main/postgresql.conf* on debian 10 for example or for Windows `C:\wapt\waptserver\pgsql`*version*`_data\postgresql.conf`), you need to increase the following parameter to approximately 1/4 the number of active WAPT agents.

```
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
```

CHAPITRE 19

Configuration des dépôts WAPT

19.1 Répliquer un dépôt

19.1.1 Aperçu fonctionnel

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

The deprecated and **unsupported** Syncthing method may be used for the Discovery versions of 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.

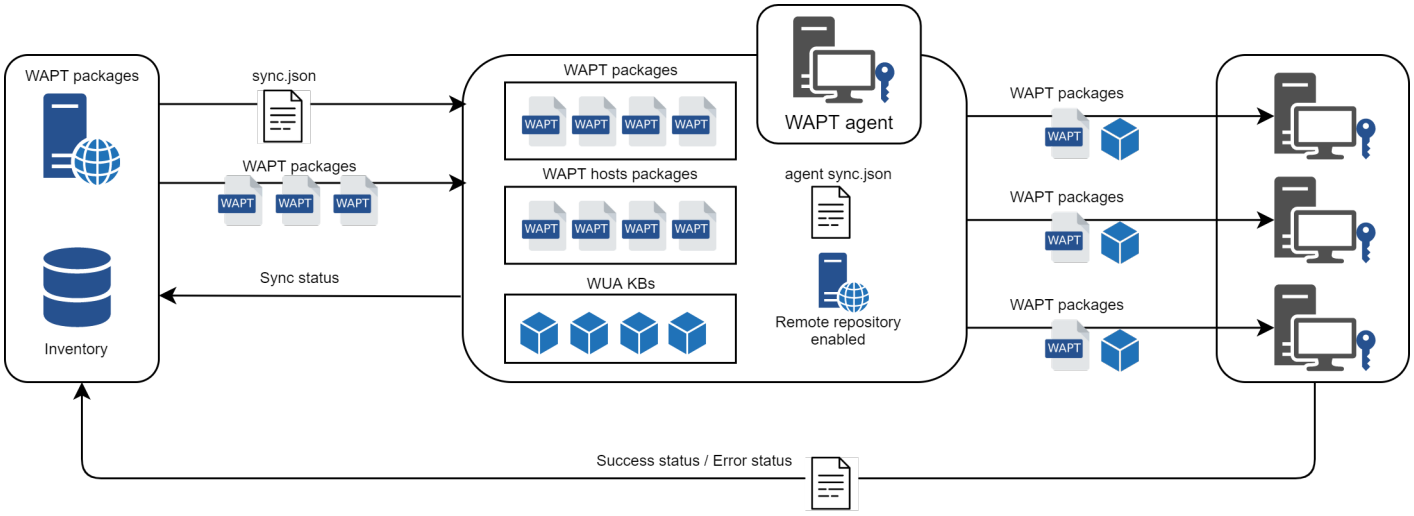


FIG. 1 – Comportement de réplication des agents WAPT

19.1.2 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	Exemple	Définition
`enable_remote_repo` = False	Permet au dépôt secondaire de se synchroniser avec le dépôt principal.	`enable_remote_repo` = True
`local_repo_path` = Dossier de WAPT /repository	**Sets the path to the root directory of the local repository** for WAPT packages.	`local_repo_path` = /var/www/
`local_repo_time_for_sync_start` = None	Définit l'heure de début de la synchronisation (HH :MM / format 24h)	`local_repo_time_for_sync_start` = 22 :30
`local_repo_time_for_sync_end` = None	Définit l'heure d'arrêt de la synchronisation (HH :MM / format 24h)	`local_repo_time_for_sync_end` = 05 :30
`local_repo_sync_task_period` = None	Définit la périodicité de la synchronisation (minutes)	`local_repo_sync_task_period` = 25
`local_repo_limit_bandwidth` = None	Définit la largeur de bande autorisée pour la synchronisation (Mbits/s)	`local_repo_limit_bandwidth` = 2
`remote_repo_dirs` = wapt,waptwua	Définit les dossiers à synchroniser	`remote_repo_dirs` = wapt,waptwua,wapt-host
`use_repo_rules` = False	Activer pour l'utilisation des *règles du dépôt*	`use_repo_rules` = True

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

Note :

A ready-to-use WAPT package
is available in **Tranquil IT public store** to enable repository replication on Windows or Linux based WAPT agents.

Ce paquet spécial :

- Installe et active le serveur web **`Nginx`** sur le dépôt seconfaire.
- 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.

Below is an example of `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)

(suite de la page précédente)

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

19.1.3 Configuration du serveur WAPT

By default, the WAPT server will know which WAPT agents are configured as remote repositories and it will list them in the WAPT console.

19.1.4 Règles du dépôt

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

The `rules.json` file is a signed *`.json`* file that contains a list of sorted rules to apply to the remote WAPT agents, so they may connect to their most appropriate repositories.

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`.

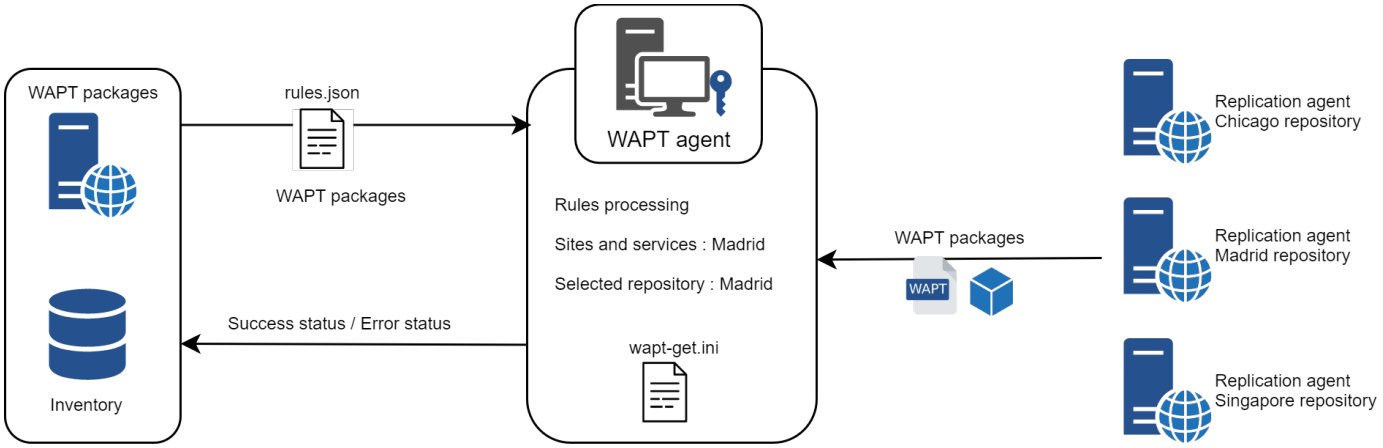


FIG. 2 – Comportement de réplication des agents WAPT

Agent WAPT

Avertissement : If you have configured GeoIP redirects on Nginx, you should disable it as it might conflict with repository rules.

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
`use_repo_rules` = 0	Pour l'utilisation de *réplication du dépôt*.	`use_repo_rules` = True

Below is an example of `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
`remote_repo_support`	`True`	Permet l'utilisation du dépôt

Console WAPT

Repository rules can be managed from the WAPT console and are based on several parameters :

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

In *Repositories*, click on the *Add rule* button. The following window appears.

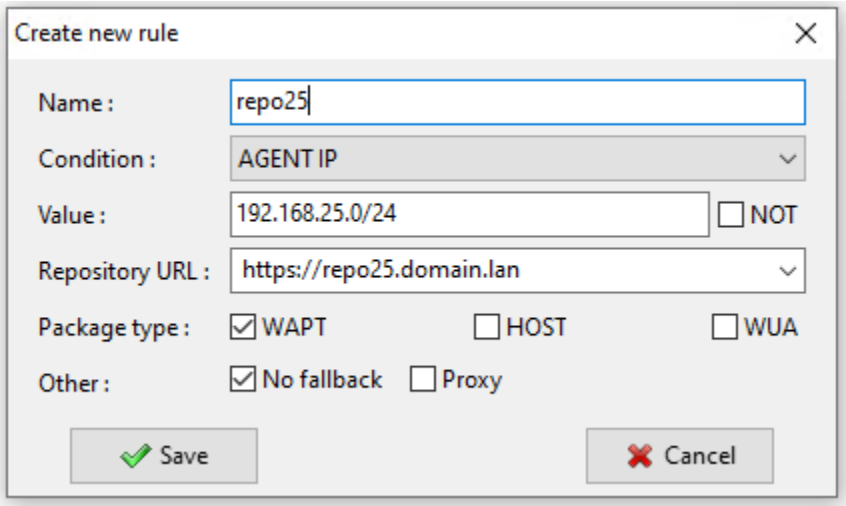


FIG. 3 – Création d'une nouvelle règle de dépôt

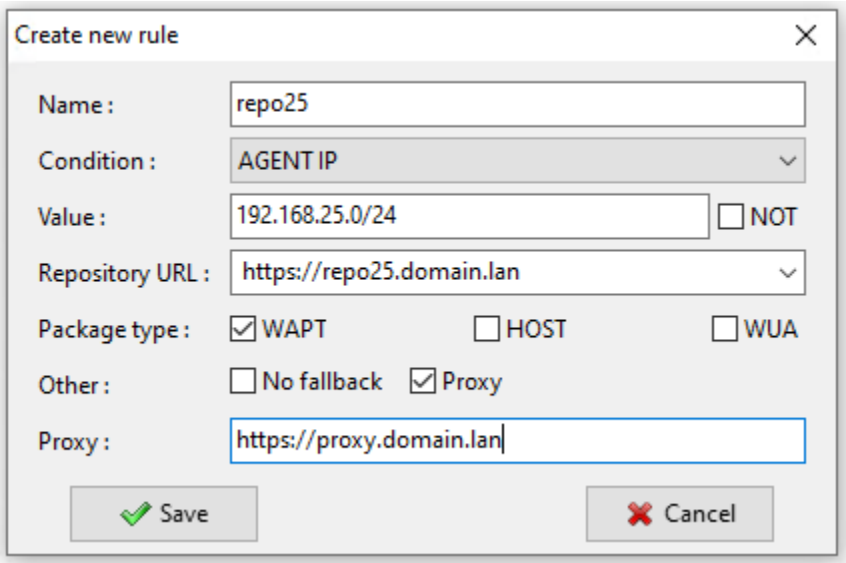
TABLEAU 3 – Détail des valeurs

Options	Exemple de valeur	Description
Name	repo25	Définit le nom de la règle.
Condition	IP de l'agent	Définit la condition à remplir pour que la règle s'applique (voir ci-dessus).
Valeur	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.
URL du dépôt	https://repo25.domain.lan	Définit la liste des dépôts secondaires disponibles. La liste inclut `http://download.windowsupdate.com/microsoftupdate/v6/wsusscan/` pour permettre le téléchargement direct des mises à jour de Windows par les dépôts secondaires afin de préserver la bande passante.
Type de paquet	WAPT	Définit quels *types de paquets* sont répliqués.
Autre	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.



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

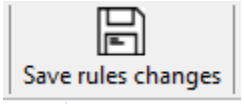


FIG. 4 – Sauvegarde de la configuration

19.2 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.

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

19.2.1 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
`http_proxy` = None	Définit l'adresse du proxy HTTP.	`http_proxy` = http://user:pwd@host_fqdn:port
`repo_url` = None	Définit l'adresse du dépôt WAPT principal.	`repo_url` = https://srvwapt.mydomain.lan/wapt
`timeout` = None	Définit le délai d'attente lors de la connexion à des dépôts distants.	`timeout` = 5
`use_http_proxy_for_repo` = False	Définit si un proxy doit être défini pour accéder aux dépôts.	`use_http_proxy_for_repo` = 1
`verify_cert` = None	Définit si *Les certificats HTTPS du dépôt doivent être vérifiés*, et si c'est le cas définit le chemin vers le paquet de certificats.	`verify_cert = C:\Program Files (x86)\wapt\ssl\server\srvwapt.mydomain.lan.crt` (sur Windows)

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.

19.2.2 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
client_certificate = None	Définit le dossier qui contient les certificats utilisés pour authentifier les paquets externes téléchargés.	client_certificate = C:\Program Files (x86)\wapt\ssl\server\srvwapt.mydomain.lan.crt (sur Windows)
client_private_key = None	Définit le dossier qui contient la clé privée.	client_private_key = C:\Program Files (x86)\wapt\ssl\server\srvwapt.mydomain.lan.pem (sur Windows)
http_proxy = None	Définit l'adresse du proxy à utiliser pour accéder au dépôt externe référencé dans la [section].	http_proxy = http://proxy.mydomain.lan:8080
public_certs_dir =	Définit le dossier qui contient les certificats utilisés pour authentifier les paquets externes téléchargés.	public_certs_dir = C:\private
repo_url = None	Définit l'adresse du dépôt WAPT externe.	repo_url = http://srvwapt.mydomain.lan/wapt
timeout = 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.	timeout = 2
verify_cert = None	Pour la *vérification des certificats HTTPS*.	verify_cert = 1

CHAPITRE 20

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.

20.1 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.

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.

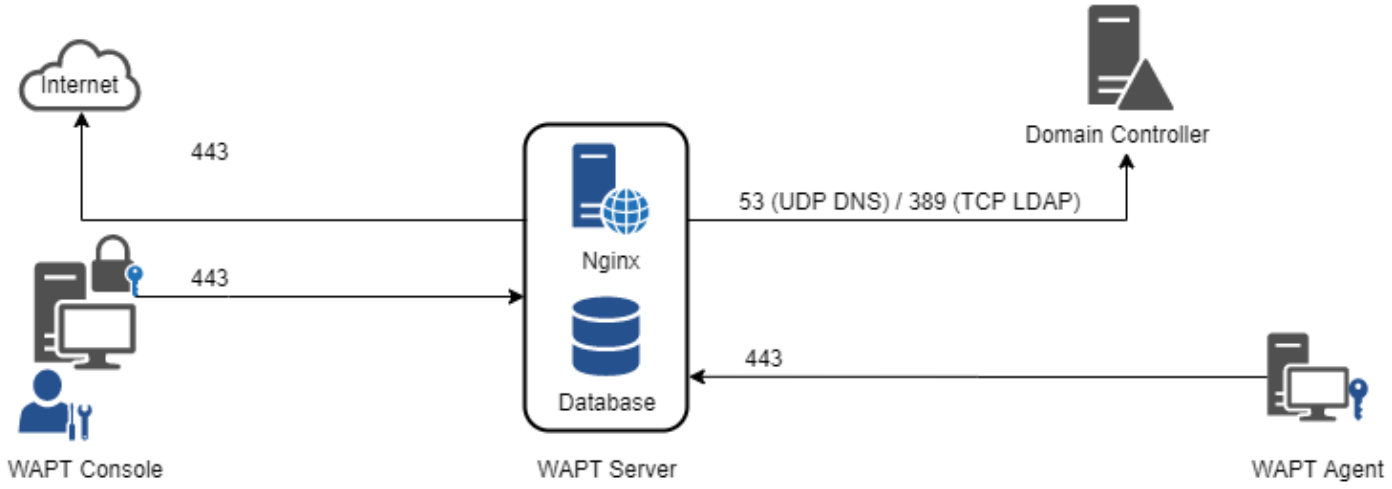


FIG. 1 – Diagramme de flux de données de WAPT

20.1.1 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
```

20.1.2 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
```

20.2 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.

20.2.1 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_CONTROLER>* avec le nom de votre contrôleur de domaine (par exemple : **srvads.mydomain.lan**).

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

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

20.2.2 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.

20.2.3 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*.

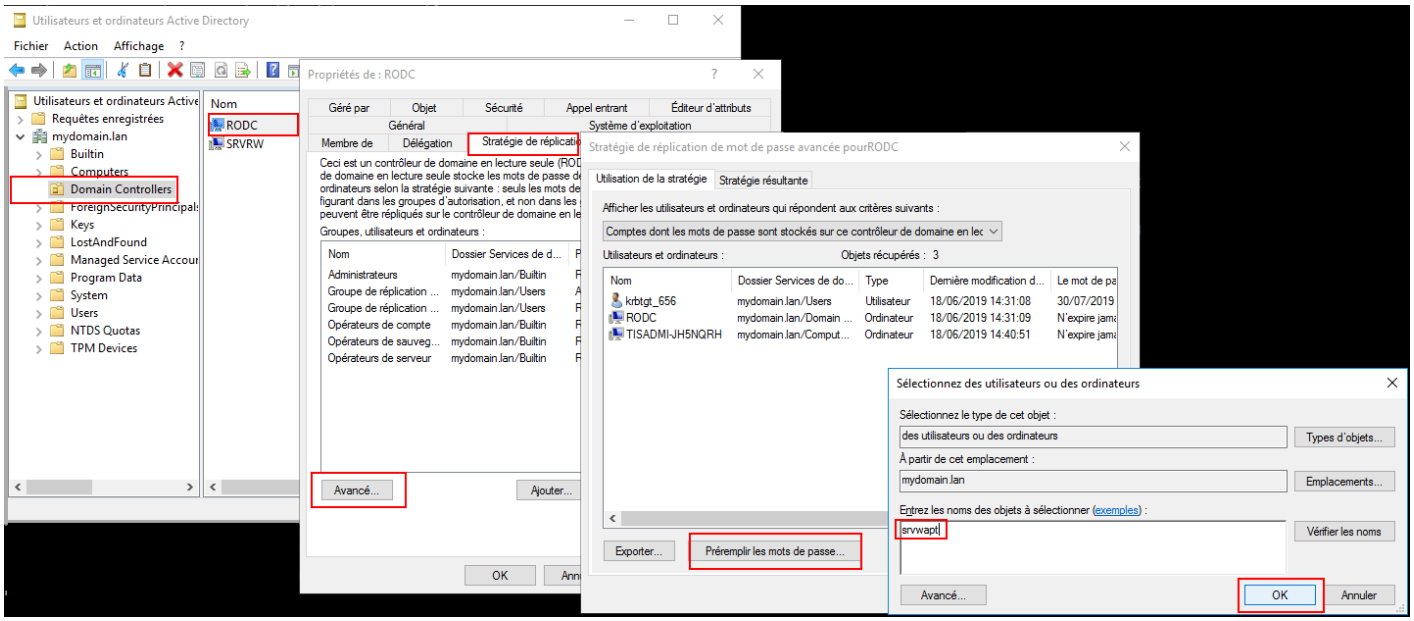
— Transférez ce fichier dans `/etc/nginx/` (avec **`winscp`** par exemple).
— Appliquez les droits d'accès appropriés au fichier `http-krb5.keytab`.

```
    # 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
```

WAPT agents only have access to a RODC domain controller

— Pour RODC (Read-Only Domain Controller), ajoutez le compte *srvwapt* au groupe de mots de passe autorisés pour la réplication.
— 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
```

20.2.4 Débugger les problèmes avec les kerberos

Attention :

- The WAPT server address cannot be an IP, Kerberos works well only with 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

To switch to a system account you must use the **psexe** tool from Microsoft : `psexe`.

— In **cmd** as an Administrator.

```
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

— On the Linux machine.

```
[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
[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

— On the Linux machine.

```
      [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,127.0.01/8,192.168.0.0/16,10.0.0.0/8,172.16.
```

(suite sur la page suivante)

(suite de la page précédente)

```
↪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):
     * 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
```

(suite sur la page suivante)

(suite de la page précédente)

```
      * 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,127.0.01/8,192.168.0.0/16,10.0.0.0/8,172.16.
↪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␣
↪YIIGagYGKwYBBQUCoIIGXjCCBlqgDTALBgkqhkiG9xIBAgKiggZHBIIGQ2CCBj8GCSqGSIb3EgECAgEAboIGLjCCBiqgAwIBBaEDAgEOogc
↪+lGapoxK1hCC62Mnye9zF4SzOEKBDLPD77KJp2xbW227lc5ZF/
↪22wGXn8n6Sw1xndf1brq1mSEUo0TzpFUfY1wNoRDaw7WUNhQK2nbTTeMrsiiPACuQtG82W0VrZJZ+4z1Gq3ZFTLYUrlC010S1T8pNRzCLFR
↪OBq9EQd+i/2Mfp8XWy46gGRtezTk8Dya+SH3henhB+L7G4ew0c3MKxFRkv0nXQ65qPAXWyogbivI/
↪ReekU1anHLnGfDyfeBw2QUM8t2kEEcSBmNKfrQ1U/u1jnlRZJ1o067PiziZsH7w/
↪zGpe7uhOa8a4RKYu1LeJEuU+CKrufulQWKuqdiwIBdq9ApoQqduCbNsE9ihH1srL3RYh9XdQ4Unx51eo49nZQA+c2Aj4JvCafbY/
↪jeRBw6SNYCrfGETN1mXytjLRyVBtJlch7djBGUAYaH1HGNfEVt+VnCW4O9OoqgC0M++u6d7Ci5w494ZseNXnF7RBKr01aVEt0231geGglNv
↪arutH6c3CzLb+xPMAOUtCIup0M43SR0DC7gJ/
↪xZ3BZyKHF6b3p3tAWiByat2XNMxfMbgjaiv7oLCNEIAga0IgTg5f0nlahTI9323vfIH8aLNNVYVJeFKNGX1ord0YpJ3RLDshNBnoDTPyCKn
↪QXiFQAezmRut+hfXtoVch8LHC0OIaloDUk1eHlFbAqQ98OaE3SZ8Fx4m8Nw3JgQ0E+zXPt8DJCnNY4YV4j3+9b1093XhsJRyP97qEFazUGF
↪PzqFnxSHtikXjCnjtzfuHLePMWn0HKFbL/hEMmAnfZ15JiBgfBi82OXv3rCui+GKT/ZsjUFsgR8tCUJ58/gXBu9J/
↪gY1R46CvWTnl03+2JHQyomm6k0XnAU+s2hX+n/QcKbIjT7ew/f/
↪UuT0J1YV+bQ8MMTpQqbau4f5sVaembIB7hTVyttpfbBEqCOVk39xZ/r8b9CMpmukShPgeJI0x353i7bO9/
↪mBkchFaeyOc45jA7Z4iJ3IHNiWLaWyYLktH/1N9/dXas8/CoZK0UsKjm9+xlkFFSP18CFHijILlIc1sTdMnAil/
↪jwqQl1W2WRnBlt8r/yE56EDR/i8VkCMHT5XsiiMCHm4LldkmDnIo/+GgHTG+3Z78Pkq939rMati/
↪gzd9geYM8aUuYwJpcb53YsjWvJD1gDEHEwS3K1MYxbyy9eiODCOCgvIeKmVPouNrugXs4TX6PJsCQDtzusSWxmZY4820HxmJkNT1lG5Zkkt
↪b93cbD4oaOWlbPViBpwLZ+TTCkeGAxo3eBicENHSK81EIJYMBfEWTTsjYPEPsl5BK7IFcqArfEWG6HQDw3bOfYAB1ZJb0zSbhyD/
↪rKnRmtSke/
↪eWIAjYaeHDX0qYMruJCuI2lYofHtFwMEKSB1jCB06ADAgESooHLBIHIDVUTdDas6nA0obxBuM2bQiZ0ZUPhAVGMoTniuCmBXU/
↪mFRAsD029zDjfl0nzeFsPdC4UBERcc8Vh4r3YeZIxUxzn5tXCW4oFypYi5kHAdx6Zd4GkZcEpzAhRF7JwSylerZiCF+fnSIiI5wdDG56PMF
      > 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␣
↪oYG3MIG0oAMKAQChCwYJKoZIhvcSAQICooGfBIGcYIGZBgkqhkiG9xIBAgICAG+BiTCBhqADAgEFoQMCAQ+iejB4oAMCARKicQRvQoZWpMI
↪x0oFJX6n4DnhPZxrq/RnjwkoTnik7R8MJkKRuvYncBfTGBIHvTJktq6+j9pHqmBDH5D5L8A
< WWW-Authenticate: Basic realm=""
< Cache-Control: store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
<
* Closing connection 0
```

(suite sur la page suivante)

(suite de la page précédente)

```
kerberos connection seems to be working
```

Verifying that you are successfully obtaining a Kerberos ticket

Attention : Exécutez toujours les commandes dans le compte système (voir le point précédent) !

```
klist purge
klist get http/srvwapt.ad.mydomain.lan
```

You must get (in your language) :

```
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
                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 0x80000
                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, vérifier 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 : You must first configure Firefox for kerberos authentication.

— Type **about:config** in the URL bar in your Firefox.
— Edit `network.negotiate-auth.trusted-uris`, and add the url of the WAPT Server : `srvwapt.mydomain.lan`.
— Vous pouvez maintenant visiter l'url : 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`.
— Reboot the host you are testing with and re-run the keytab creation process again.

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.

20.3 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 `verify_cert`	Fonctionnement de l'agent WAPT
`verify_cert` = 0	the WAPT agent will not check the WAPT Server HTTPS certificate.
`verify_cert` = 1	the WAPT agent will check the WAPT Server HTTPS certificate using the certificate bundle. `C:\Program Files (x86)\wapt\lib\site-packages\certifi\cacert.pem`
`verify_cert` = C :\Program Files (x86)\wapt\ssl\srvwapt.mydomain.lan.crt	the WAPT agent will check the WAPT Server HTTPS certificate with the certificate bundle. `C:\Program Files (x86)\wapt\ssl\srvwapt.mydomain.lan.crt`

Indication : To quickly and easily enable verification of the HTTPS certificate, you can use the *Pinning* method.

20.3.1 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 :

If *wapt-get enable-check-certificate* returns an error,
remove the *.crt* with same name on `C:\Program Files (x86)\wapt\sslserver`

Note :

- The command **enable-check-certificate** downloads the certificate `srvwapt.mydomain.lan.crt` in the folder `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\srvwapt.mydomain.lan.crt`.
- L'agent WAPT va maintenant vérifier les certificats en utilisant le certificat épinglé.

Attention : If you use the *certificate pinning* method, **BE REMINDED** to archive the `/opt/wapt/waptserver/ssl` folder on your WAPT Server.

The file will have to be restored on your server if you migrate or upgrade your WAPT Server, if you want the WAPT agents to continue to be able to establish trusted HTTPS connections with the WAPT Server.

20.3.2 How to use a commercial certificate or certificates provided by your Organization ?

Si la méthode d'épinglage ne vous convient pas, vous pouvez remplacer le certificat auto-signé généré lors de l'installation de **WAPT**.

Replace the old certificate with the new one in the folder `/opt/wapt/waptserver/ssl/` (Linux) or `C:\wapt\waptserver\ssl\` (Windows).

The new key pair must be in PEM encoded Base64 format.

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 srvwapt.mydomain.lan.crt ca.crt > cert.pem`

— **For Linux servers it is also necessary to reset**
the 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.
— Use a WAPT package to modify `wapt-get.ini` and push the certificate.

20.3.3 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:\Usersadmin\AppData\Local\waptconsole\waptconsole.ini`.

Ceci définit correctement l'attribut `verify_cert` pour la communication HTTPS entre la console WAPT et le serveur WAPT.

20.4 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.

On large and security-minded networks, the SuperAdmin account should not be used since it cannot provide the necessary traceability for administrative actions that are done on the network assets.

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.
- However, all actions on the WAPT equipped remote devices are based on X.509 signatures, so an *Administrator* will need both an Active Directory login **AND** a private key whose certificate is recognized by the remote devices that the Administrator manages using 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).

20.4.1 Activation de l'authentification Active Directory

- To enable authentication of the WAPT Server with Active Directory, configure the file `waptserver.ini` as follows.

Note :

The WAPT Server configuration file on GNU/ Linux and macOS systems
is found in `/opt/wapt/conf/waptserver.ini` or in `/opt/wapt/waptserver/waptserver.ini`.

The WAPT Server configuration file on Windows systems
is found in `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.

20.4.2 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 : The WAPT Server uses the Certificate Authority *bundles* from the operating system for validating the SSL/ TLS connection to 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
```

(suite sur la page suivante)

(suite de la page précédente)

```
    # Windows
certutil -addstore -f "ROOT" cainterne.crt
```

— Une fois que vous avez configuré le LDAP SSL/ TLS sur votre Active Directory (veuillez vous référer à la documentation Microsoft pour cela), vous pouvez activer la prise en charge de la sécurité SSL/ TLS pour AD dans `waptserver.ini`.

```
ldap_auth_ssl_enabled = True
```

— Redémarrer le service **`waptserver`**.

20.5 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 :

We strongly recommend enabling Kerberos or login/password registration
in the WAPT server post-configuration.

Avertissement : Toutes les actions sont à mener sur le serveur WAPT

20.5.1 Activation de l'authentification des certificats côté client

Avertissement : Pour **Linux**, vérifiez si le lien symbolique dans `sites-enabled` existe :

```
cd /etc/nginx/sites-enabled/
find . -maxdepth 1 -type l -ls
```

The expected result should be :

```
269091      0 lrwxrwxrwx   1 root     root           36 juil. 22 15:51 ./wapt.conf -> /etc/nginx/
↪sites-available/wapt.conf
```

Otherwise use the following command :

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

To enable the authentication, you need to add those parameters to the following configuration file in the *location* section :

— Sur Linux : `/etc/nginx/sites-available/wapt.conf`.

— Sur Windows : `C:\wapt\waptserver\nginx\conf\nginx.conf`.

```
location / {
...
  proxy_set_header X-Ssl-Authenticated $ssl_client_verify;
  proxy_set_header X-Ssl-Client-DN $ssl_client_s_dn;
  if ($ssl_client_verify != SUCCESS) {
      return 401;
  }
...
}
```

Attention :

Please note that as of 2024-01-09, WAPT does not support
CRL, which means that when you delete a machine in the WAPT console, the machine will still have access to the WAPT repository.

Avertissement :

WAPTDeploy cannot use https to retrieve the WAPT agent,
you will have to add this section in the file :

```
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;
}
```

20.6 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.

20.6.1 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.

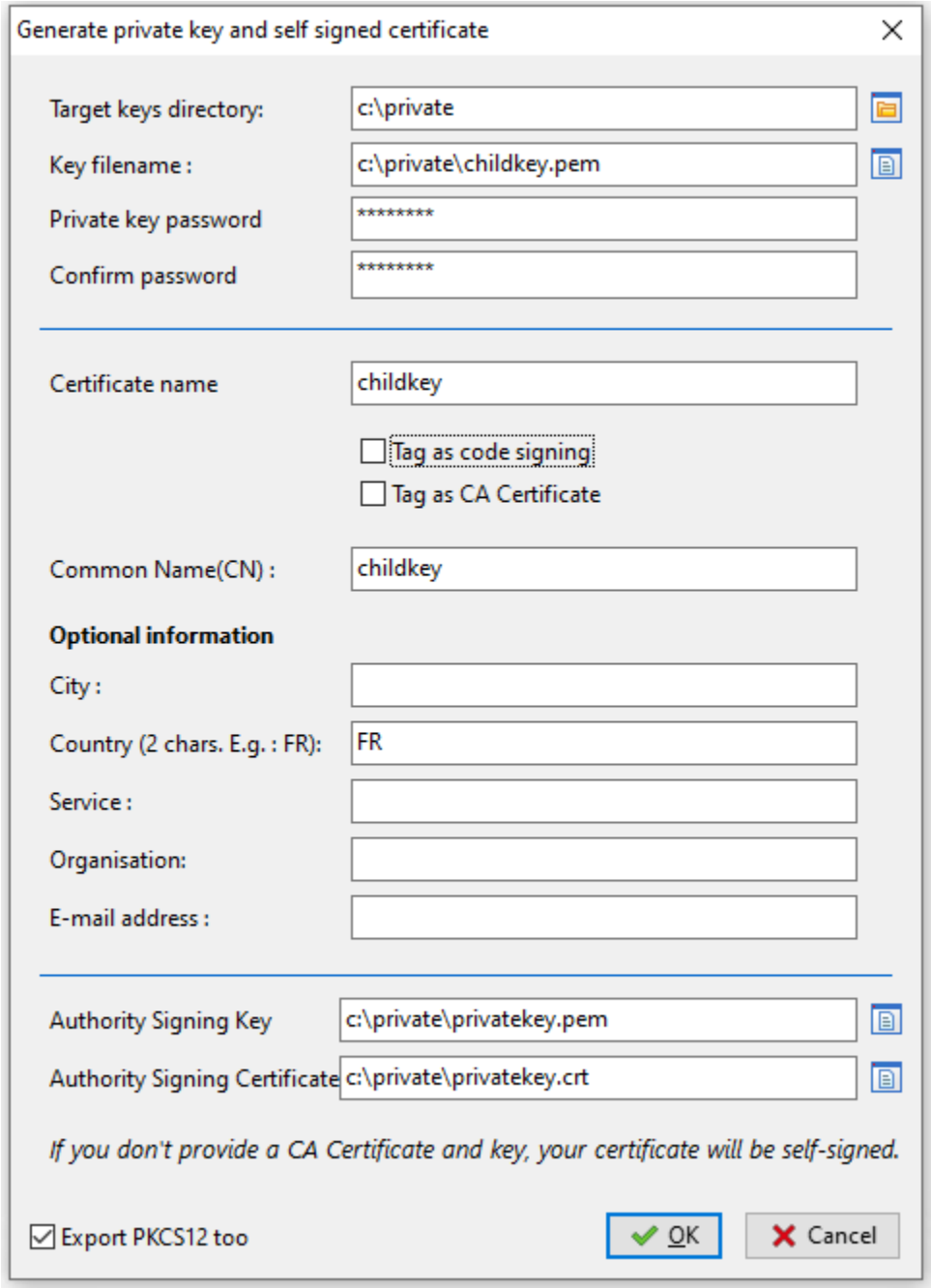


FIG. 2 – Génération d'un certificat sans l'attribut *Pour Signature de code*

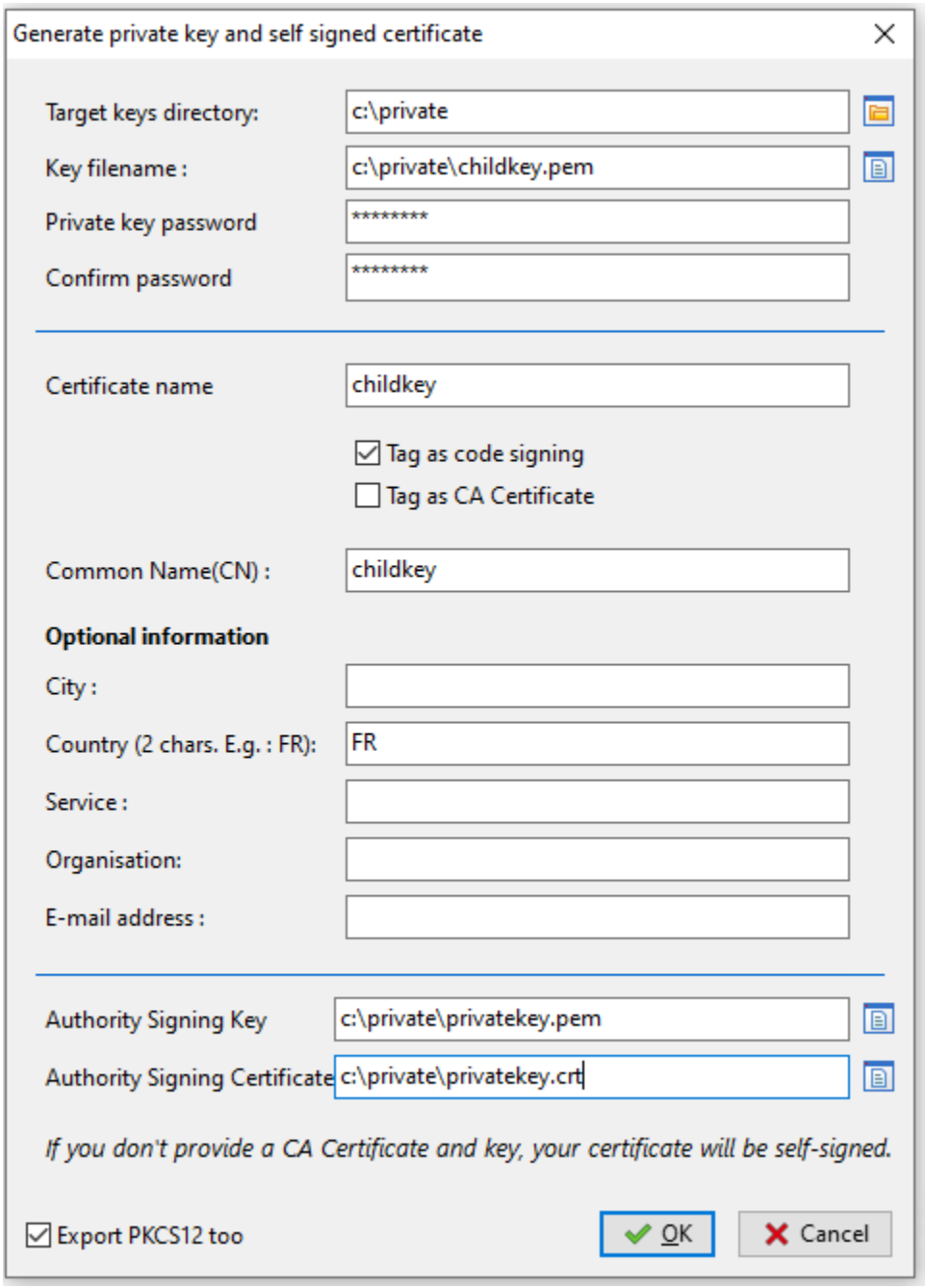


FIG. 3 – Génération d'un certificat avec l'attribut *Pour Signature de code*

20.7 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*

20.8 Configuration des listes de contrôle d'accès

Indication :

Default `admin` user of WAPT are authenticated by password
stored in `waptserver.ini` as a value of `wapt_password` key.

Others WAPT users may be local users (`htpasswd_path`)
or AD account users (`ldap_auth_server` / `ldap_auth_base_dn`).

Les ACL définissent les actions autorisées pour tous les types d'utilisateurs dans le contexte WAPT.

Note :

Default ACLs user level are defined by `default_ldap_users_acls`
in `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.

ACLs simply limit what actions the server is allowed to relay from the WAPT console to the WAPT agents.

A la date du |date|, les agents WAPT ne vérifient pas les droits ACL.

To configure ACLs in WAPT, go to *Tools* → *Manage WAPT users and rights*.

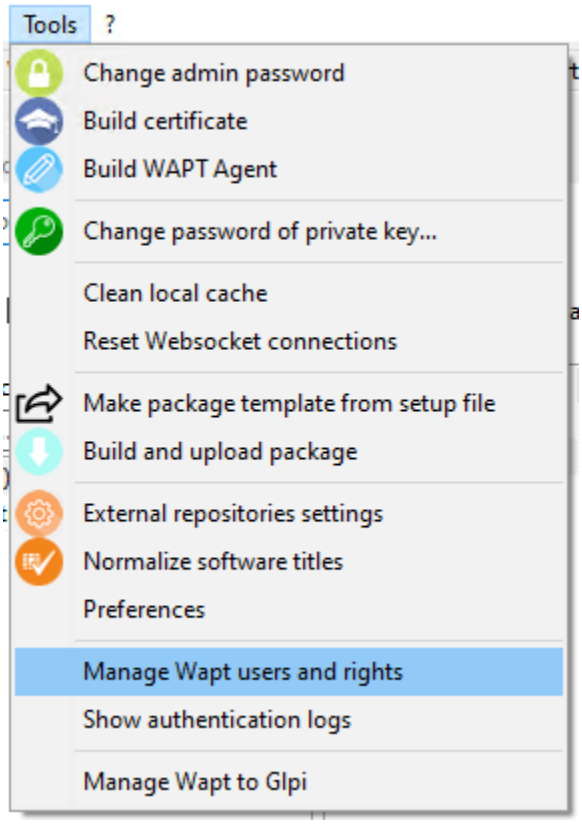


FIG. 4 – Gérer les utilisateurs et les droits WAPT dans l'onglet Outils

Au premier lancement après l'installation du serveur, seul le compte *SuperAdmin* est présent dans la liste des utilisateurs.

Note :

If the *SuperAdmin* account does not exist or does not have the *admin* right,
then the account is recreated by restarting the waptserver service.

The *SuperAdmin* account is authenticated using the value of `wapt_password`
in the `waptserver.ini` configuration file.

Deux types de comptes sont gérables par ACL, *local* et *Active Directory*.

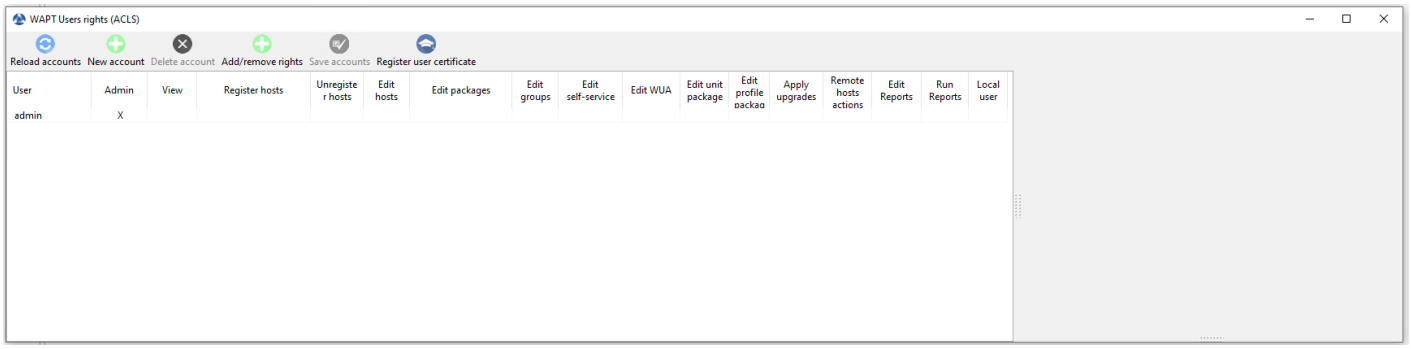


FIG. 5 – Gérer les utilisateurs et les droits WAPT au premier lancement

20.8.1 Compte d'utilisateur local

Les utilisateurs locaux sont définis par un fichier *.htpasswd*.

Configuration du serveur WAPT

For using local user accounts, you need create a file named `waptusers.htpasswd` in the same *folder* on the WAPT server containing the `waptserver.ini` file.

— Create the `waptusers.htpasswd`.
  — On Linux :

```
touch /opt/wapt/conf/waptusers.htpasswd
chown wapt  /opt/wapt/conf/waptusers.htpasswd
```

  — On 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*.

  It is possible to rename accounts by pressing F2
  on the *User* column.

— Sauvegardez en cliquant sur *Enregistrer les comptes*.
— Pour définir un mot de passe, voir le point **Changez le mot de passe**.
— For setting rights, see the section on *managing ACL rights*.

If the local user has a password in `waptusers.htpasswd`, then the username appears in **bold** and *Local User* is checked, else change the password for this user.

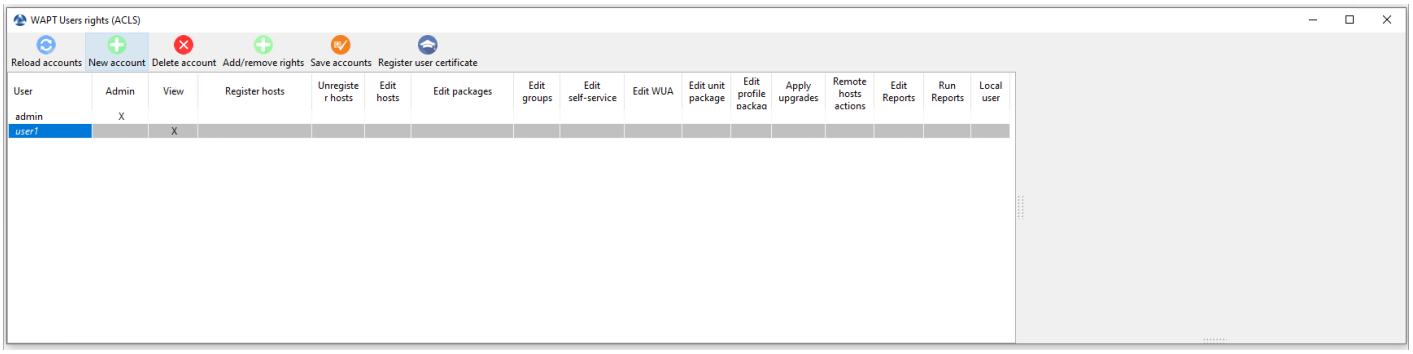


FIG. 6 – Création d'un nouveau compte local

Changer le mot de passe de l'utilisateur

Pour changer le mot de passe du compte sélectionné :

- Faites un :menuselection `:clic droit sur le compte --> Changer le mot de passe utilisateur sur le serveur Wapt`.
- Saisissez le nouveau mot de passe.

L'utilisateur local apparaît en *gras* et la case *Mots de passe* est cochée.

20.8.2 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.

20.8.3 Blocage des comptes d'utilisateurs locaux

Pour désenregistrer les utilisateurs locaux, faites :menuselection `: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.

20.8.4 Liste des droits

De nombreux *droits et restrictions* peuvent être définis pour chaque utilisateur dans la console WAPT.

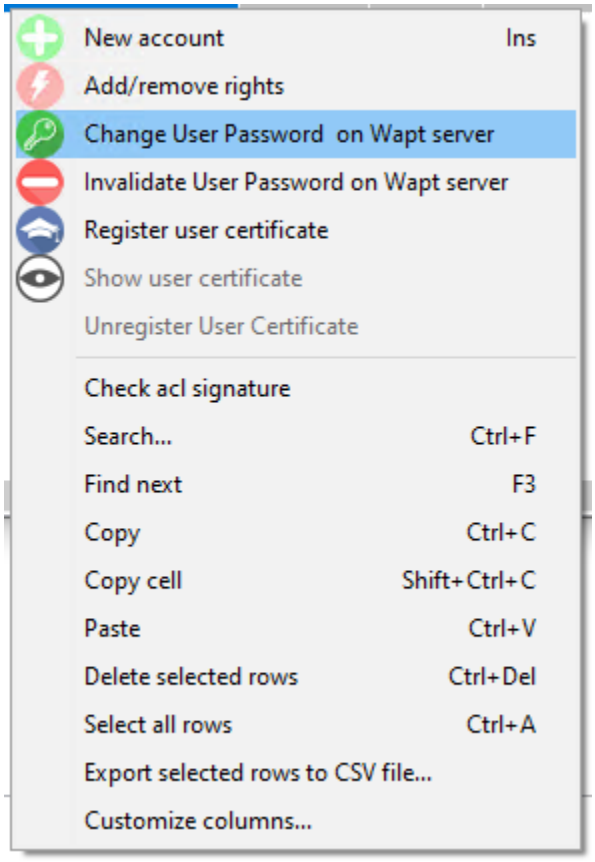


FIG. 7 – Changer le mot de passe de l'utilisateur

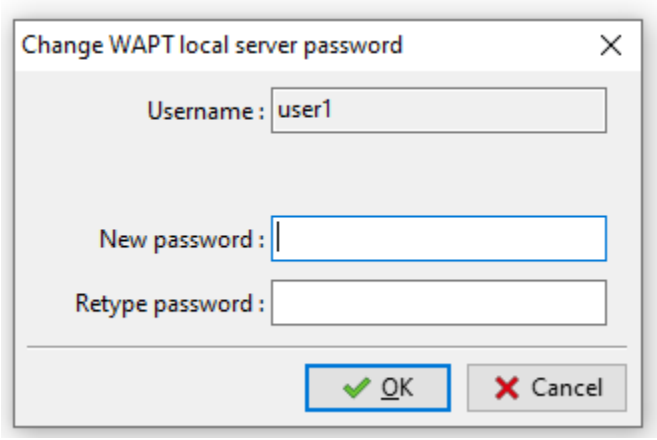


FIG. 8 – Saisir le nouveau mot de passe

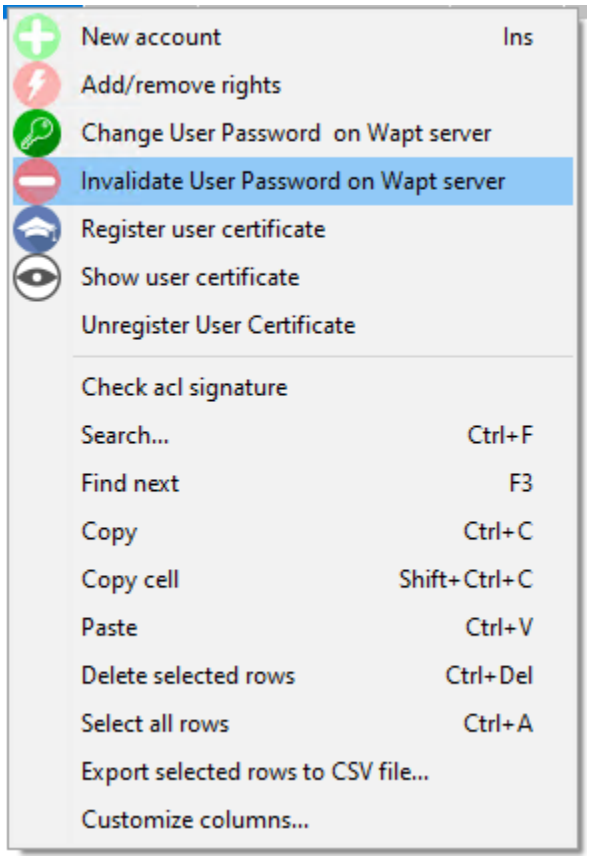


FIG. 9 – Invalidation d'un compte d'utilisateur local

Tableau 3 – Liste des droits des utilisateurs

Droit	Description
Admin	Comme SuperAdmin, tous les droits sont accordés sauf *Mot de passe*.
Voir	Permet de visualiser uniquement les informations sur la console WAPT.
Inscrire machine	Allows to use the Admin credentials to *register manually a host* with the WAPT server.
Désinscrire machine	Permet de *supprimer une machine* depuis la console WAPT.
Modif machine	Permet de *modifier le paquet machine* sur la console WAPT.
Modif paquets	Permet de *modifier les paquets de base* qu'elle est autorisée à modifier.
Modif groupes	Permet de *modifier les paquets de groupe* sur la console WAPT.
Modif self-service	**Allows to *modify self-service rules*** on the WAPT console.
Modif WUA	**Allows to *modify WUA / WSUS rules*** on the WAPT console.
Modif paquets AD OU	Permet de *modifier les paquets unit* sur la console WAPT.
Modif paquets Profile	**Allows to *modify profiles packages*** on the WAPT console.
Lancer les instalations	**Allows to remotely apply upgrades on her perimeter of hosts,** if host is on **PENDING** status.
Actions distantes machine	Allows to make use of *Windows Computer Management tool* with the WAPT console.
Modifier requêtes	Permet de *créer ou modifier des requêtes de rapport*.
Lancer requête	Permet de *exécuter des rapports SQL existants*.
Mot de passe	Définit un utilisateur local

20.8.5 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 :
- **Updated packages are available only to computers** where certificates are deployed.

— **ACL are valid only on the perimeter of hosts**
where the certificates are deployed.

Associer un certificat à un utilisateur

Indication : Par défaut, aucun certificat n'est défini pour aucun utilisateur (y compris *SuperAdmin*).

The account in the WAPT console appears in *italic*
if no certificate is associated to the user.

To associate a certificate to an user, do *Right-Click on user* → *Register user certificate*.

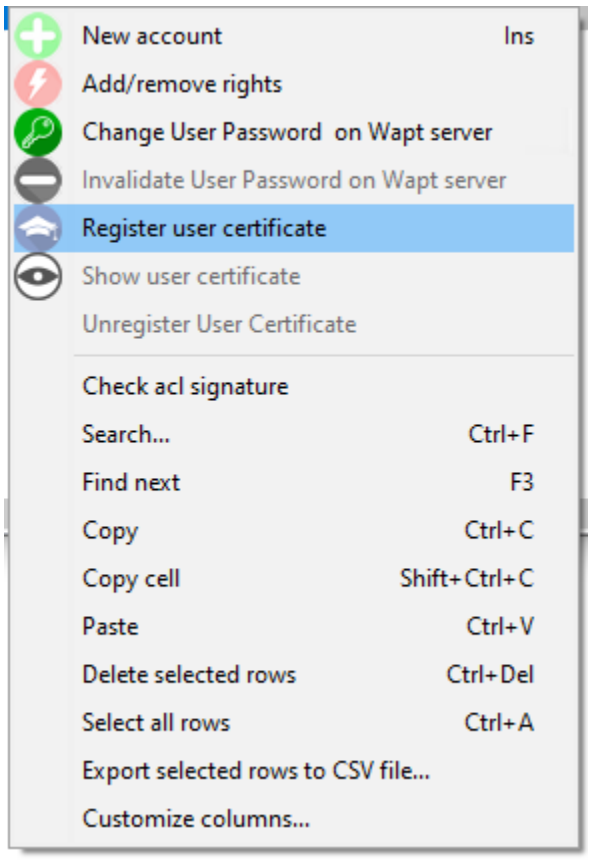


Fig. 10 – Enregistrement des certificats d'utilisateur

Ensuite, choisissez le certificat à associer à l'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 :

It is possible to do a multiple selection by using keyboard shortcuts
Crtl+left-click and pressing the `spacebar`.

Restreindre le périmètre des droits accordés à l'utilisateur

It is possible to associate a perimeter to a right given to a user.

Vue

Tableau 4 – Définition du périmètre autorisé

Périmètre	Description
Tout refuser	Aucun droit de regard n'est autorisé (non coché).
Autoriser sur tout le périmètre	La visualisation est autorisée pour tous les agents WAPT.
Autoriser des périmètres spécifiques	**View is allowed on the selected perimeter defined** as a list of certificates.
Autoriser où le certificat d'utilisateur est déployé	**View is allowed only on the perimeter where the certificate** of the Administrator is deployed.

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
Interdire tous les paquets	Aucune édition n'est autorisée pour aucun paquet (non coché).
Autoriser tous les paquets	Le droit de modification est autorisé pour tous les paquets.
Autoriser des noms de paquets spécifiques	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 :

After each server update, update your *console*
then *regenerate* the WAPT agent.

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 WAPT 1.5	Vers WAPT 1.6	Vers WAPT 1.7	Vers WAPT 1.8	Vers WAPT 2.0
Depuis WAPT 1.3	✓	✓	✗	✗	✗
Depuis WAPT 1.5	—	✓	✓	✗	✗
Depuis WAPT 1.6	—	—	✓	✗	✗
Depuis WAPT 1.7	—	—	—	✓	✗
Depuis WAPT 1.8.2	—	—	—	—	✓

CHAPITRE 21

Mise à jour mineure du serveur WAPT

21.1 Debian

Indication : This process is only valid for Debian (*Ubuntu Here*).

— Before updating, if not done already apply this *configuration* to your repositories.
— If it is correct, execute the following block.

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

— Then run the **post-configuring** script.

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

Avertissement : The post-configuration script rewrites the nginx configuration. If you use a *special configuration*, save your `wapt.conf` file with the command :

```
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 exécuté en tant que **root**.

— Exécutez le script.

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

— Cliquez sur *Oui* pour exécuter le script postconf.

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

     < yes >          < no >
```

— Choisissez un mot de passe 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 >
```

— 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 n° 1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
  — Le choix n°2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
  — Le choix n°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
(x) 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 >
```

— Indiquez 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 >
```

La post-configuration est maintenant terminée.

```
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 **Nginx** pour que *le port 80 soit redirigé en permanence vers 443*

— Once completed your server is ready.

21.2 Ubuntu

— Before updating, if not done already apply this *configuration* to your repositories.
— If it is correct, execute the following block.

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

— Then run the **post-configuring** script.

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

Avertissement : The post-configuration script rewrites the nginx configuration. If you use a *special configuration*, save your `wapt.conf` file with the command :

```
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 exécuté en tant que **root**.

— Exécutez le script.

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

— Cliquez sur *Oui* pour exécuter le script postconf.

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

     < yes >          < no >
```

— Choisissez un mot de passe 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 >
```

— 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 n° 1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
  — Le choix n°2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
  — Le choix n°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
(x) 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 >
```

— Indiquez 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 >
```

La post-configuration est maintenant terminée.

```
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 **`Nginx`** pour que *le port 80 soit redirigé en permanence vers 443*

— Once completed your server is ready.

21.3 Centos / RedHat

— Before updating, if not done already apply this *configuration* to your repositories.
— If it is correct, execute the following command.

```
yum update -y
```

— Install packages.

```
yum install tis-waptserver tis-waptsetup -y
```

— Then run the **post-configuration** script.

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

Avertissement : The post-configuration script rewrites the nginx configuration. If you use a *special configuration*, save your `wapt.conf` file with the command :

```
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 exécuté en tant que **root**.

— Exécutez le script.

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

— Cliquez sur *Oui* pour exécuter le script postconf.

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

    < yes >          < no >
```

— Choisissez un mot de passe 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 >
```

— 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 n° 1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
  — Le choix n°2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
  — Le choix n°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
(x) 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 >
```

— Indiquez 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 >
```

La post-configuration est maintenant terminée.

```
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 **Nginx** pour que *le port 80 soit redirigé en permanence vers 443*

— Once completed your server is ready.

21.4 Windows

21.4.1 Discovery

Important :

Follow this procedure for getting the right packages for the WAPT Discovery
Edition. For WAPT **Enterprise** Edition please refer to the next block.

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

WAPT Discovery will be release later. For the free version,
refer to wapt-1.8 documentation https://www.wapt.fr/en/doc-1.8/.

21.4.2 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 : The post-configuration script rewrites the nginx configuration. If you use a special configuration, save your `nginx.conf` file with :

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

It will be necessary to overwrite the configuration after
the post-configuration with the command :

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

— Choisissez la langue d'installation.

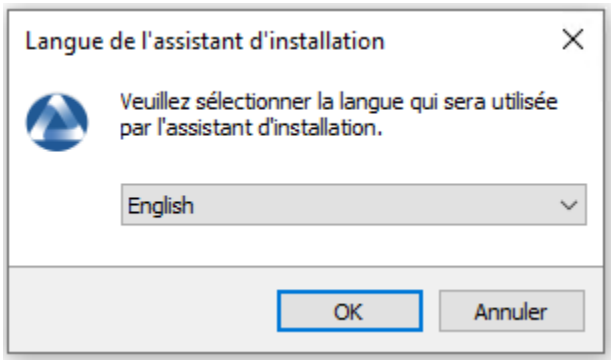


FIG. 1 – Choix de la langue pour WAPT

— 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 une tâche supplémentaire (laissez la valeur par défaut).

— Ne pas modifier le mot de passe du serveur WAPT (si cela n'est pas nécessaire).

— Ne pas créer de clé personnelle.

— Sauter la construction de l'agent WAPT, *nous le ferons plus tard*.

— Cliquez sur le bouton *Installer* pour lancer l'installation, attendez que l'installation soit terminée.

— Cliquez sur *Terminer* pour fermer la fenêtre.

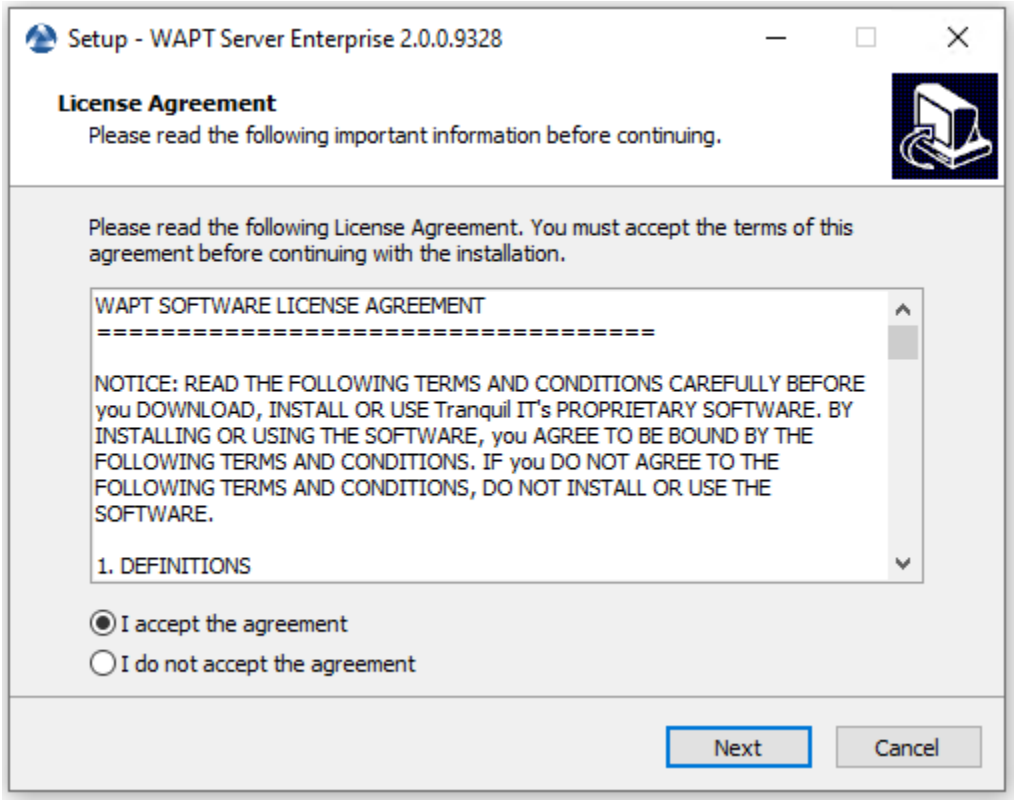


FIG. 2 – Acceptez les termes de la licence WAPT

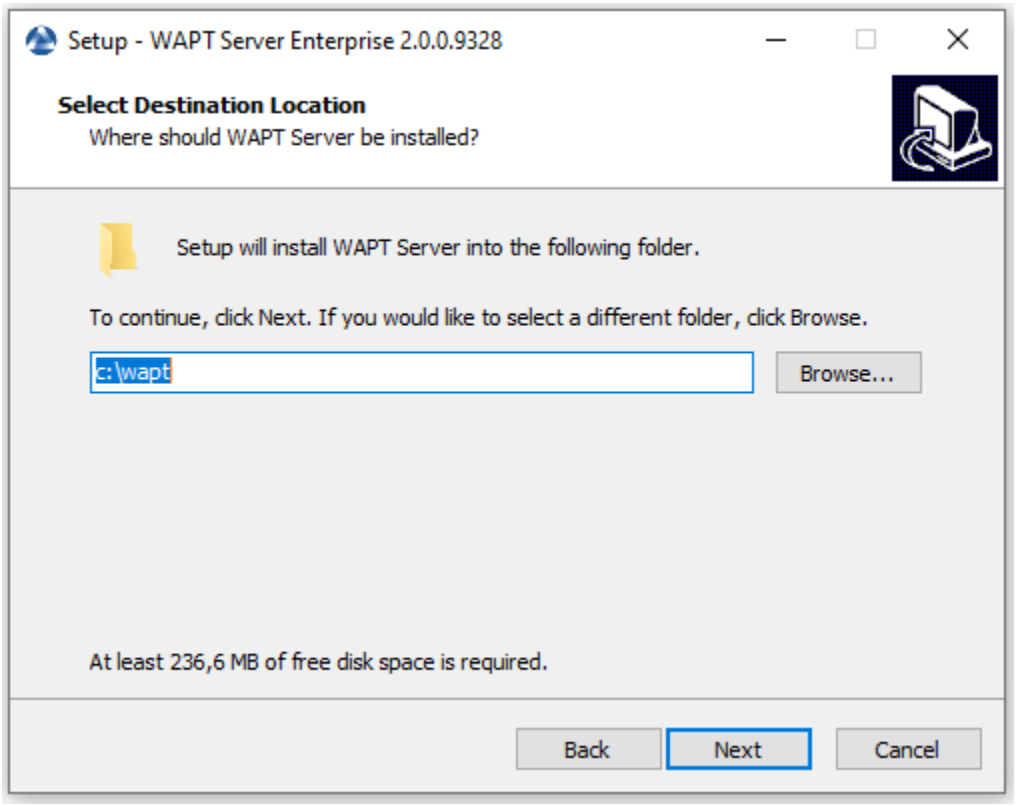


FIG. 3 – Choix du dossier de destination pour WAPT

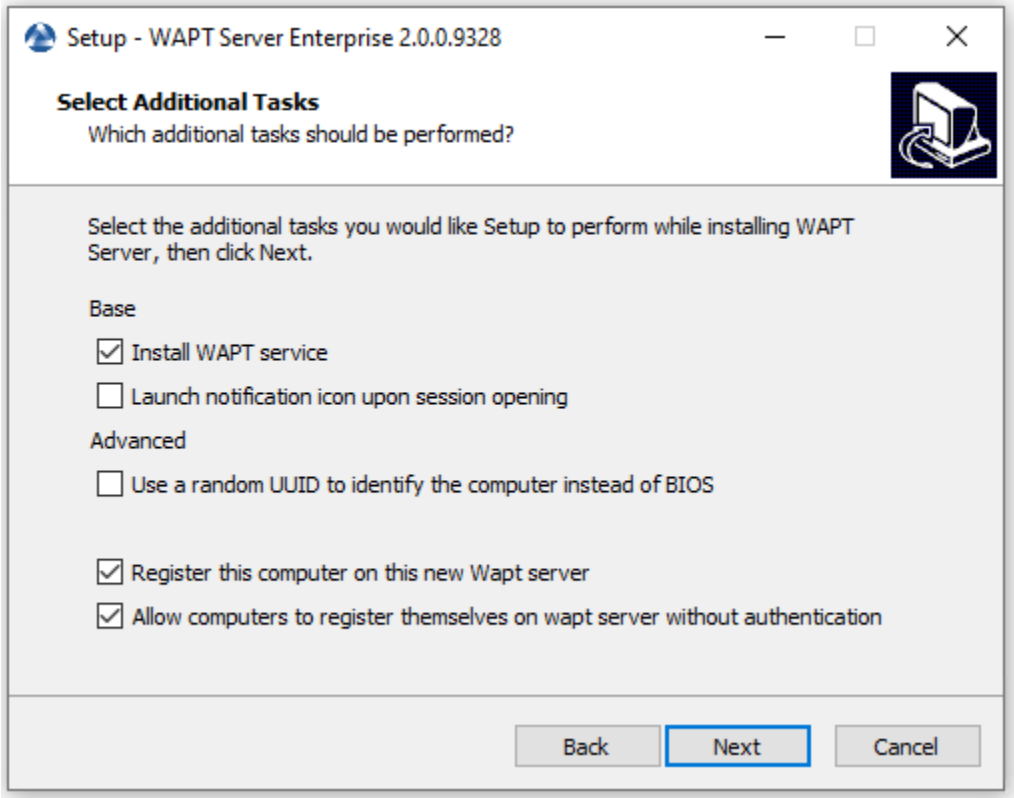


FIG. 4 – Sélection d'une tâche supplémentaire

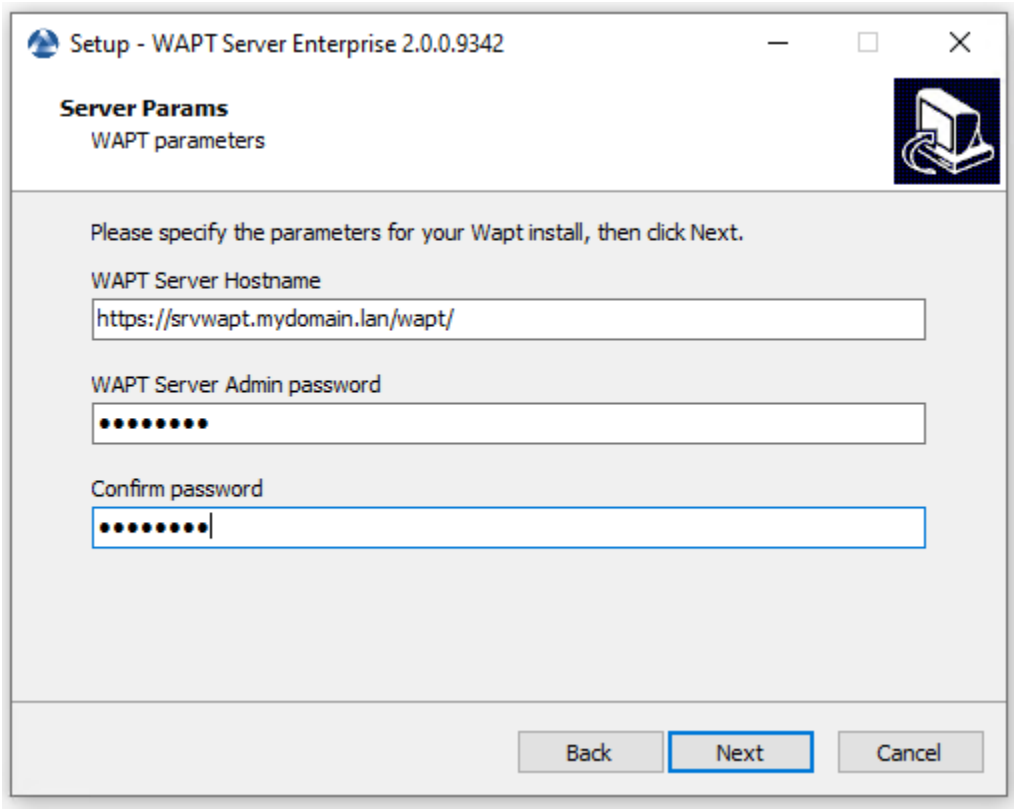


FIG. 5 – Ne pas changer le mot de passe

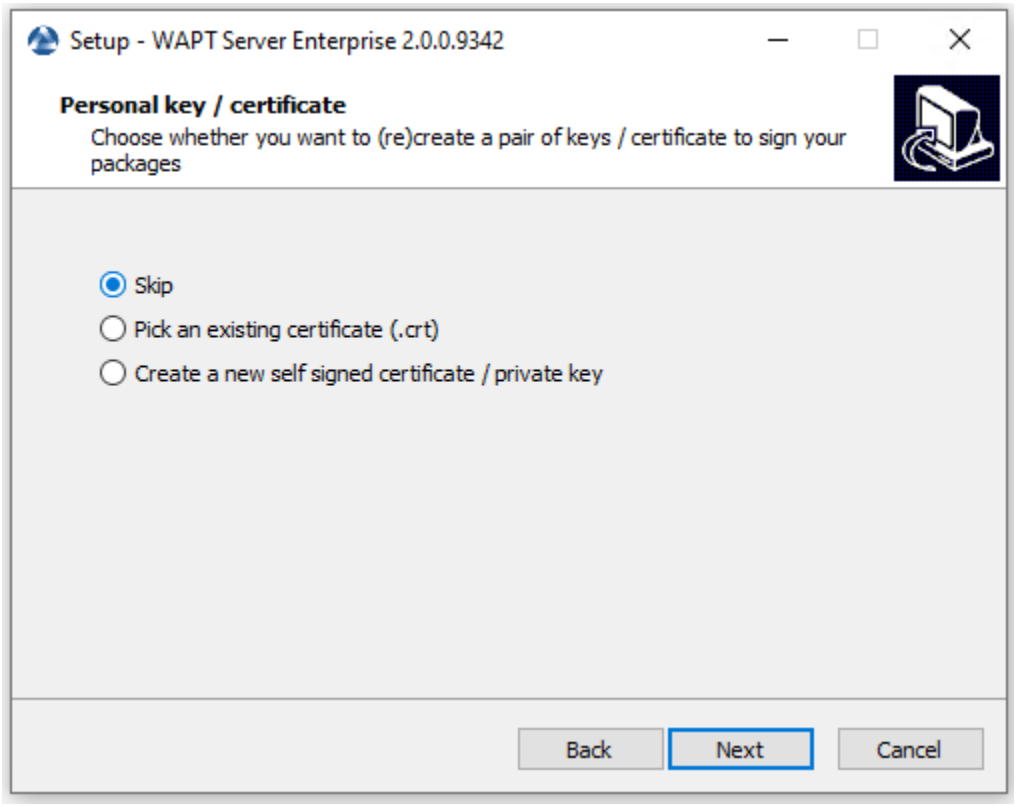


FIG. 6 – Sauter la création de la clé de signature

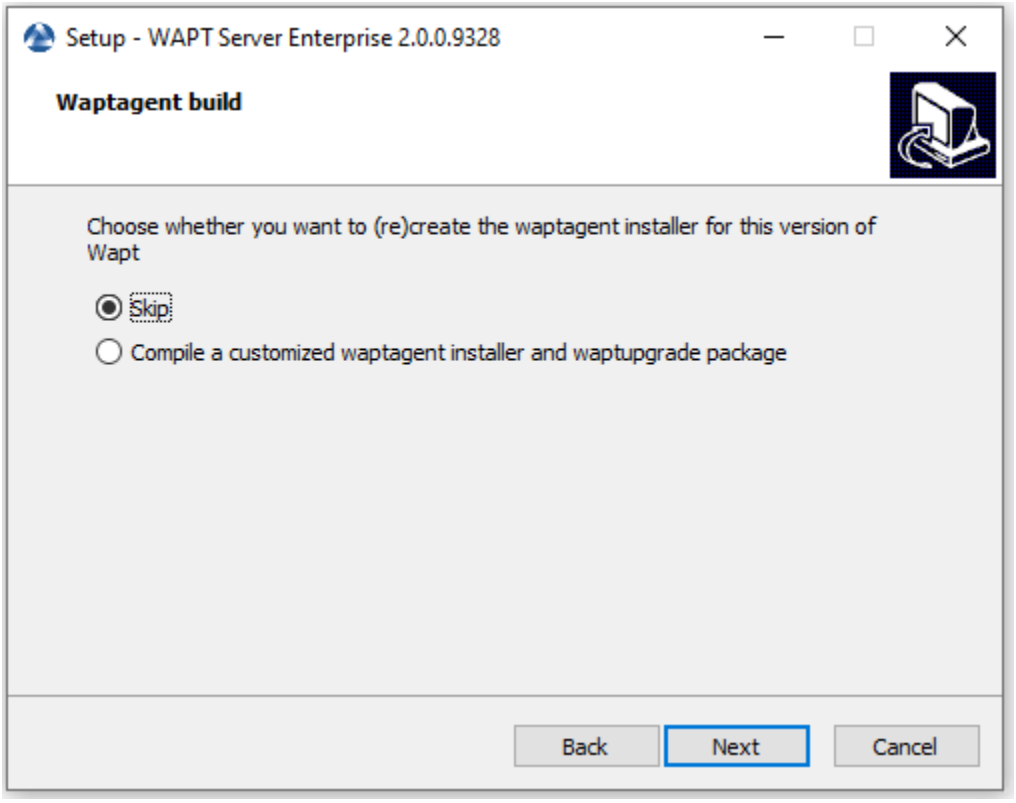


FIG. 7 – Sauter la construction de l'agent WAPT

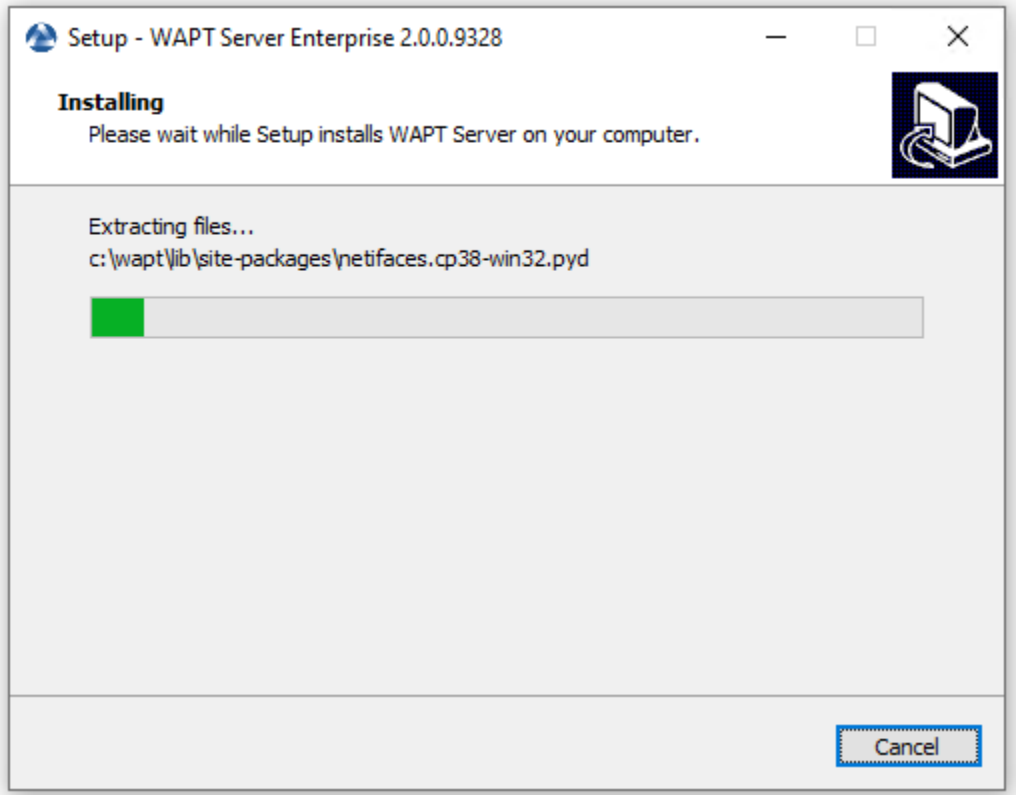


FIG. 8 – Progression de l'installation du serveur WAPT

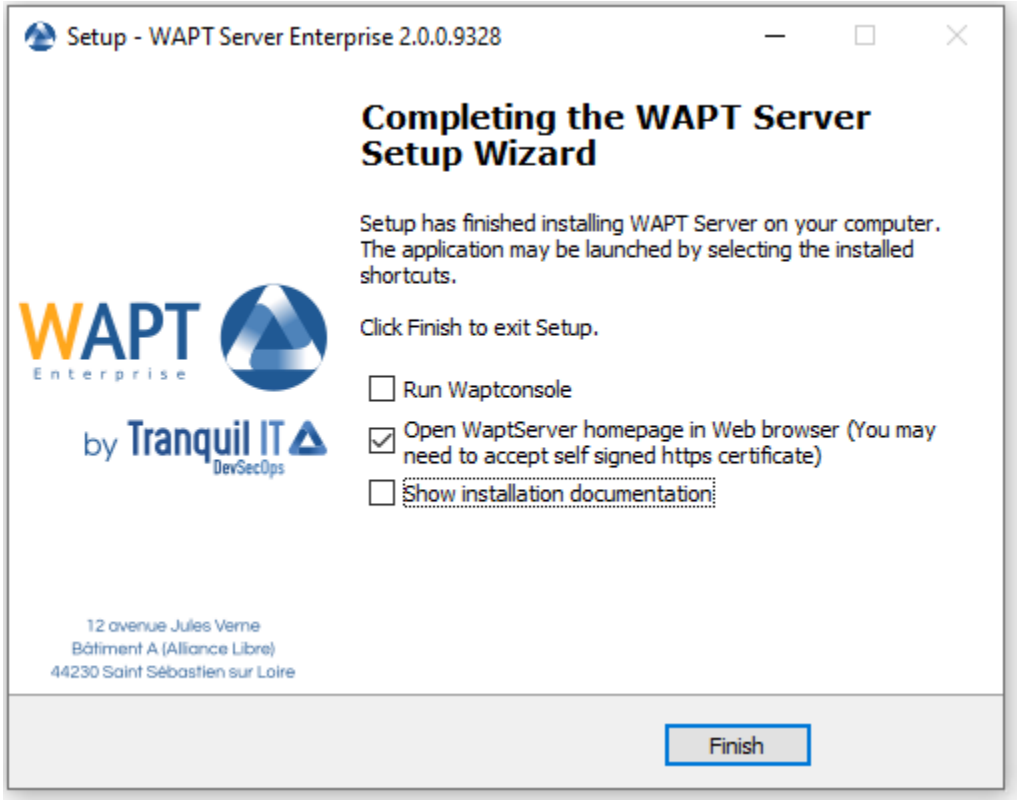


FIG. 9 – L'installation est terminée

Le serveur WAPT sur votre Windows est prêt.

CHAPITRE 22

Mise à jour de WAPT de 1.8.2 à 2.0

Note : Before upgrading, ensure that *installation requirements* are met.

Attention : Port 443 is used by the WAPT Server and must be available.

22.1 Debian

Note : If you are running on Debian9 Stretch, you have first to upgrade to Debian10 Buster before upgrading to WAPT 2.x. **WAPT-Server 2.x is not available for Debian9**.

— 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.

22.1.1 De Community à Discovery

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

WAPT Discovery will be release later. For the free version,
refer to wapt-1.8 documentation https://www.wapt.fr/en/doc-1.8/

22.1.2 Enterprise

Important :

Follow this procedure for getting the right packages for the WAPT Discovery
Edition. For WAPT **Enterprise** Edition please refer to the next block.

— Install `apt-transport-https` for the use of https.

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

— Retrieve the key *.gpg* and add it to the Tranquil IT repository.

```
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
```

— **If the file does not exist, create `wapt.conf`**
in `/etc/apt/auth.conf.d` to store your login information.

Indication :

Replace user and password to access WAPT Enterprise repository,
with those provided by our sales department.

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

— Apply the correct rights, update the repository and install the packages.

```
chmod 600 /etc/apt/auth.conf.d/wapt.conf
apt update
apt install tis-waptserver tis-waptsetup
```

22.1.3 Post-configuration

— Lancer l'étape de post-configuration.

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

Avertissement : The post-configuration script rewrites the nginx configuration. If you use a *special configuration*, save your `wapt.conf` file with the command :

```
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 exécuté en tant que **root**.

— Exécutez le script.

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

— Cliquez sur *Oui* pour exécuter le script postconf.

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

     < yes >          < no >
```

— Choisissez un mot de passe 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 >
```

— 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 n° 1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
  — Le choix n°2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).

— Le choix n°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
(x) 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 >
```

— Indiquez 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 >
```

La post-configuration est maintenant terminée.

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

(suite sur la page suivante)

(suite de la page précédente)

```
                    < OK >
```

Liste des options du script de post-configuration :

Options	Description
`--force-https`	Configurer **`Nginx`** pour que *le port 80 soit redirigé en permanence vers 443*

— Redémarrez le serveur WAPT.

```
systemctl restart waptserver nginx
```

— Make sure that the file owner is correct.

```
chown wapt:www-data -R /var/www/wapt*
```

— Re-signez tous vos paquets WAPT.
  — *Using the WAPT console*, or
  — *Using the command line*.

22.2 CentOS / RedHat

— Tout d'abord, mettez à jour la distribution.

```
yum update
```

— **Add or update the package repository for CentOS / RedHat packages,**
  import the GPG key from the repository and install the WAPT Server packages.

22.2.1 CentOS7

De Community à Discovery

Important :

Follow this procedure for getting the right packages for the WAPT Discovery
Edition. For WAPT **Enterprise** Edition please refer to the next block.

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

WAPT Discovery will be release later. For the free version,
refer to wapt-1.8 documentation https://www.wapt.fr/en/doc-1.8/.

Enterprise

Important :

Follow this procedure for getting the right packages for the WAPT Enterprise
Edition. For WAPT **Discovery** Edition please refer to the previous block.

To access WAPT Enterprise resources, you must use the username
and password provided by our sales department.

— Add the Tranquil IT repository.

```
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
```

Installation des paquets du serveur WAPT

— Retrieve the key `.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
```

22.2.2 CentOS8

De Community à Discovery

Important :

Follow this procedure for getting the right packages for the WAPT Discovery
Edition. For WAPT **Enterprise** Edition please refer to the next block.

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

WAPT Discovery will be release later. For the free version,
refer to wapt-1.8 documentation https://www.wapt.fr/en/doc-1.8/.

Enterprise

Important :

Follow this procedure for getting the right packages for the WAPT Enterprise Edition. For WAPT **Discovery** Edition please refer to the previous block.

To access WAPT Enterprise resources, you must use the username and password provided by our sales department.

— Add the Tranquil IT repository

```
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
```

Installation des paquets du serveur WAPT

— Retrieve the key `.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
```

22.2.3 Post-configuration

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

Avertissement : The post-configuration script rewrites the nginx configuration. If you use a *special configuration*, save your `wapt.conf` file with the command :

```
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 exécuté en tant que **root**.

— Exécutez le script.

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

— Cliquez sur *Oui* pour exécuter le script postconf.

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

     < yes >          < no >
```

— Choisissez un mot de passe 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 >
```

— 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 n° 1 permet d'enregistrer des ordinateurs sans authentification. Le serveur WAPT enregistre tous les ordinateurs qui demandent à être enregistrés.
  — Le choix n°2 active l'enregistrement initial basé sur kerberos (vous pouvez l'activer ultérieurement).
  — Le choix n°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
(x) 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 >
```

— Indiquez 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 >
```

La post-configuration est maintenant terminée.

```
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 **Nginx** pour que *le port 80 soit redirigé en permanence vers 443*

— Redémarrez le serveur WAPT.

```
systemctl restart waptserver nginx
```

— Make sure that the file owner is correct.

```
chown wapt:www-data -R /var/www/wapt*
```

— Re-signez tous vos paquets WAPT.
  — *Using the WAPT console*, or
  — *Using the command line*.

22.3 Windows

22.3.1 De Community à Discovery

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

22.3.2 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.

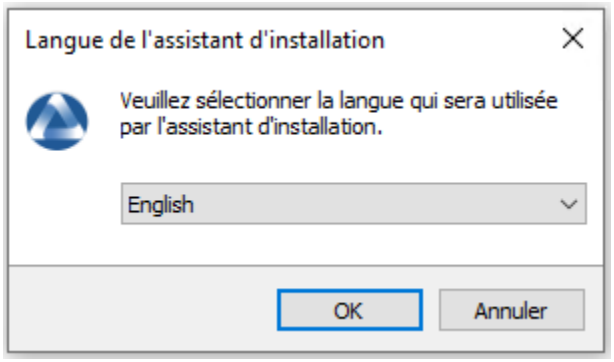


FIG. 1 – Choix de la langue pour le WAPT

— Accept the License and click on *Next* to go on to the next step.
— **Choose the installation directory (leave the default if correct)**
and click on *Next* to go on to the next step.
— **If old folder installation found, this message appear.**
Click on *Yes* to go on to the next step.
— Sélectionnez une tâche supplémentaire si nécessaire.
— Choisissez **Non** à la question de la première installation.
— Modifiez le mot de passe du serveur WAPT si nécessaire, puis appuyez sur *Suivant*.
— Ne pas créer de clé personnelle.
— Sauter la construction de l'agent WAPT, *nous choisirons de le faire plus tard*.
— Cliquez sur le bouton *Installer* pour lancer l'installation, attendez que l'installation soit terminée.
— Cliquez sur *Terminer* pour fermer la fenêtre.

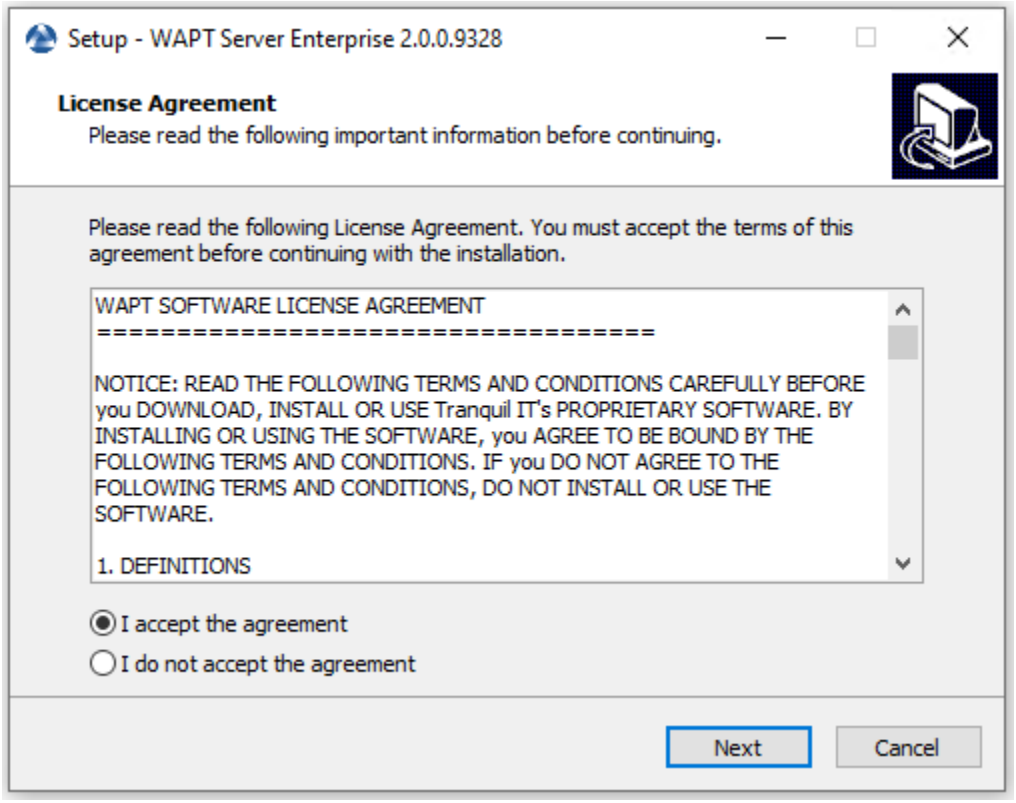


FIG. 2 – Acceptation des conditions de la licence WAPT

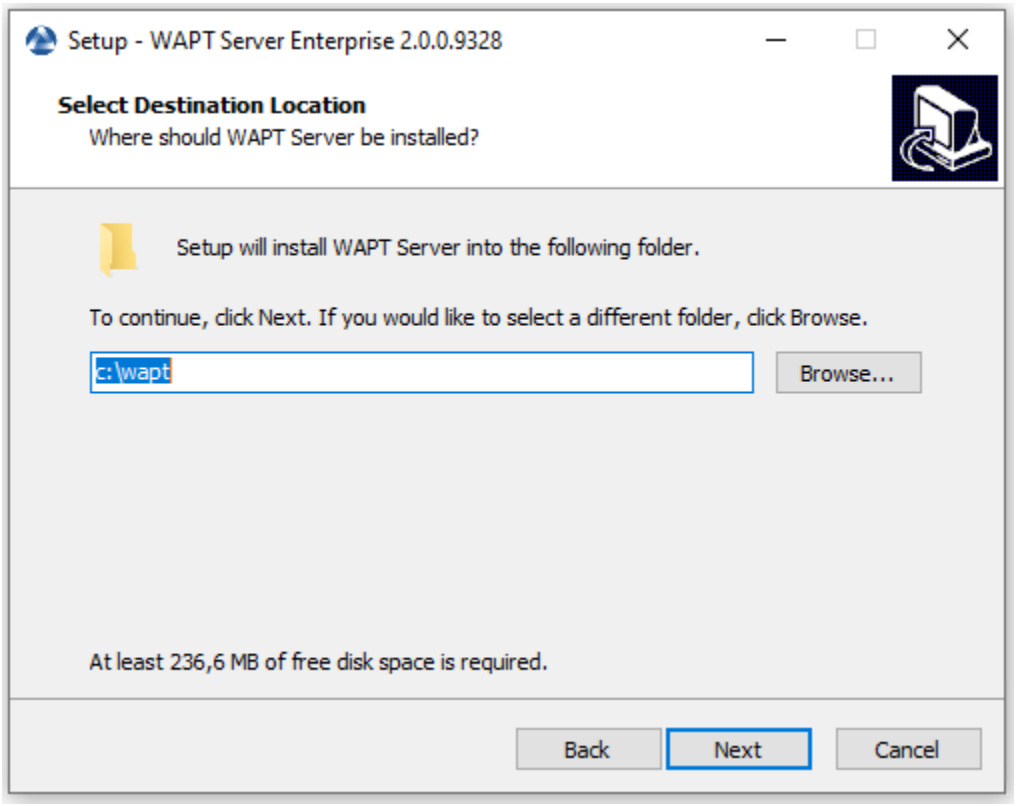


FIG. 3 – Choix du dossier de destination WAPT

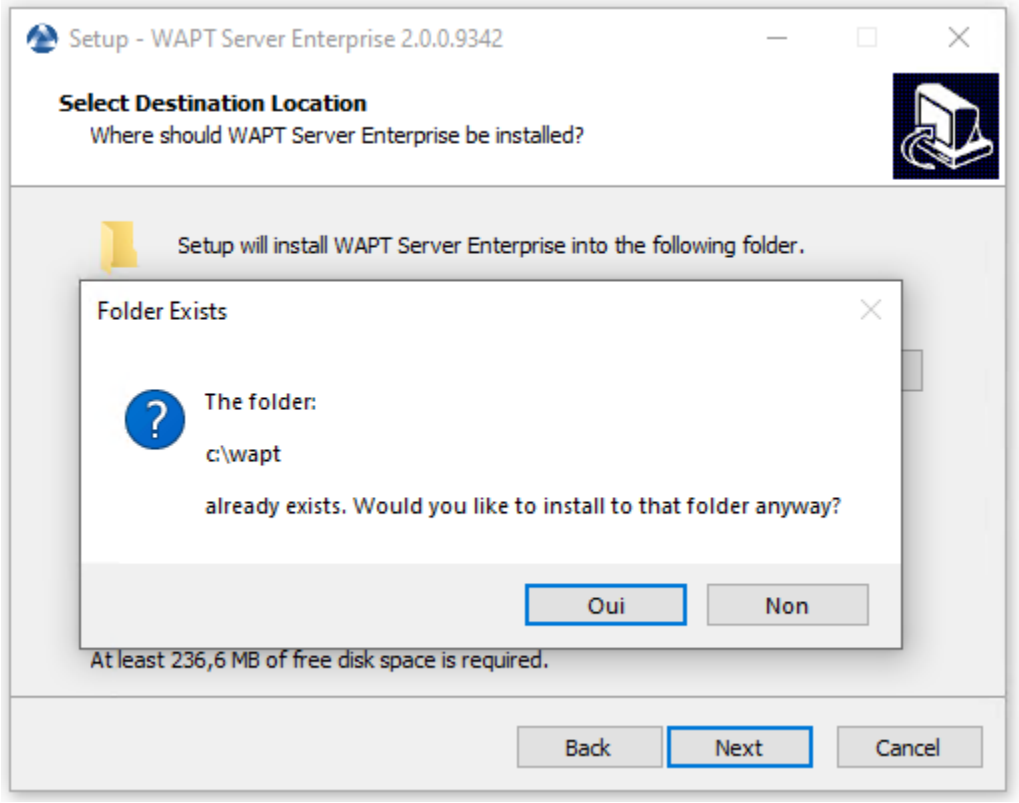


FIG. 4 – Message de l'ancien dossier Dossier de destination WAPT

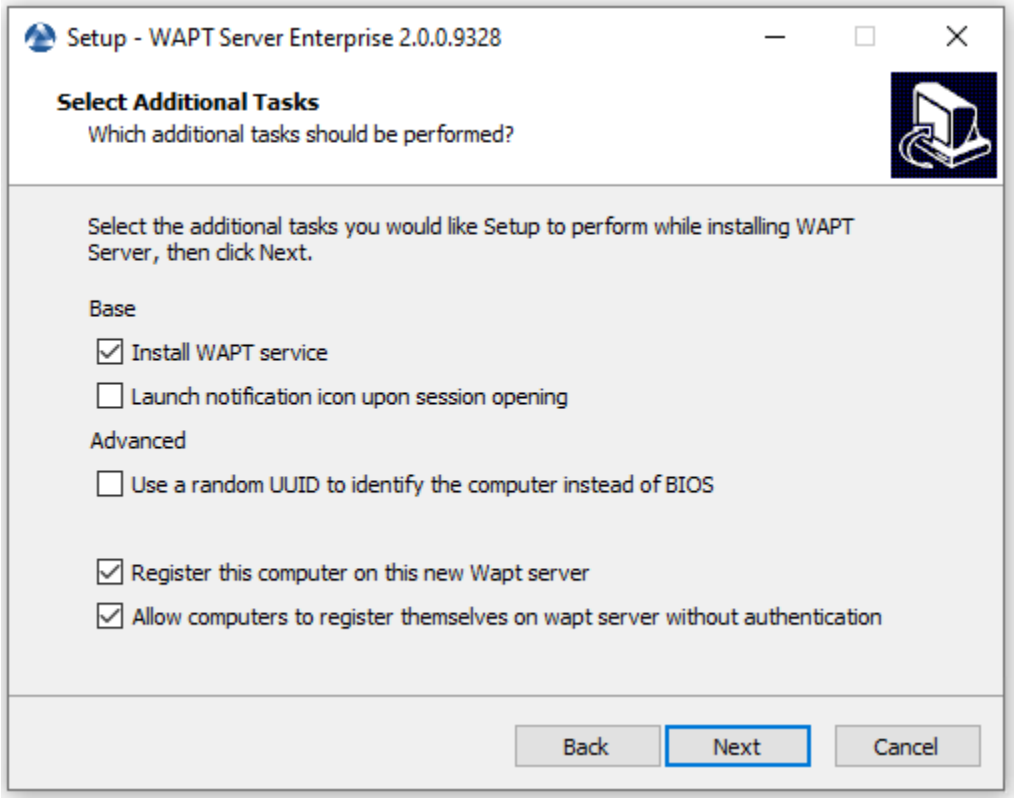


FIG. 5 – Sélection d'une tâche supplémentaire

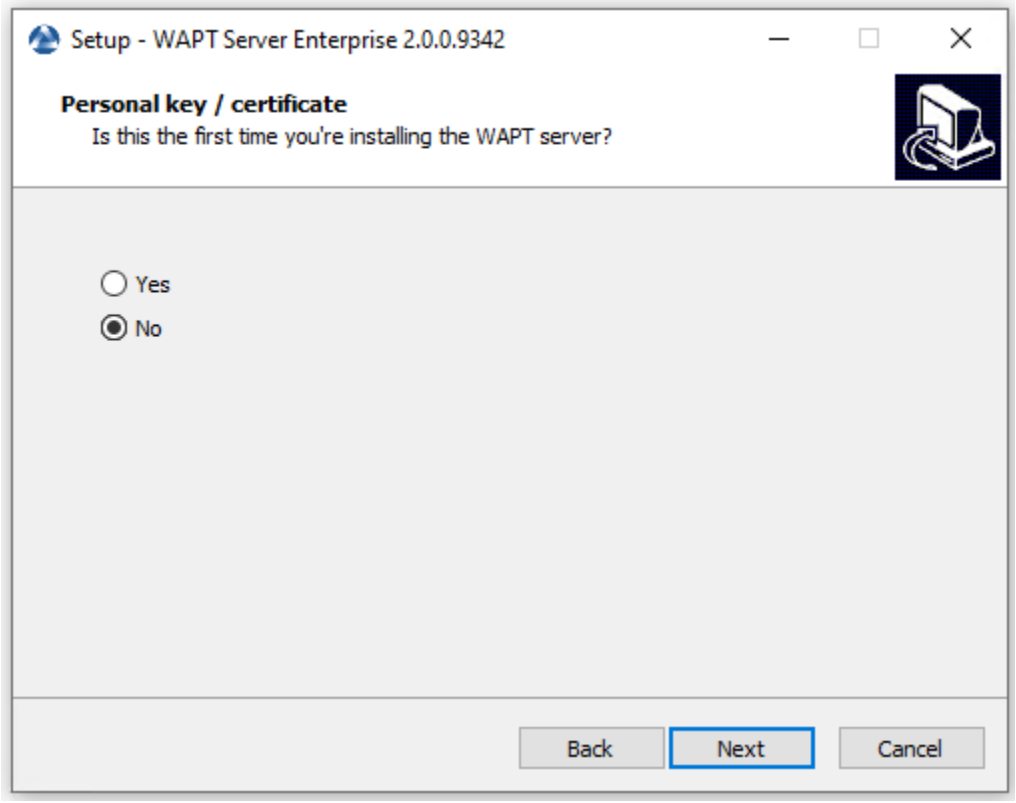


FIG. 6 – Choisissez Non à la question

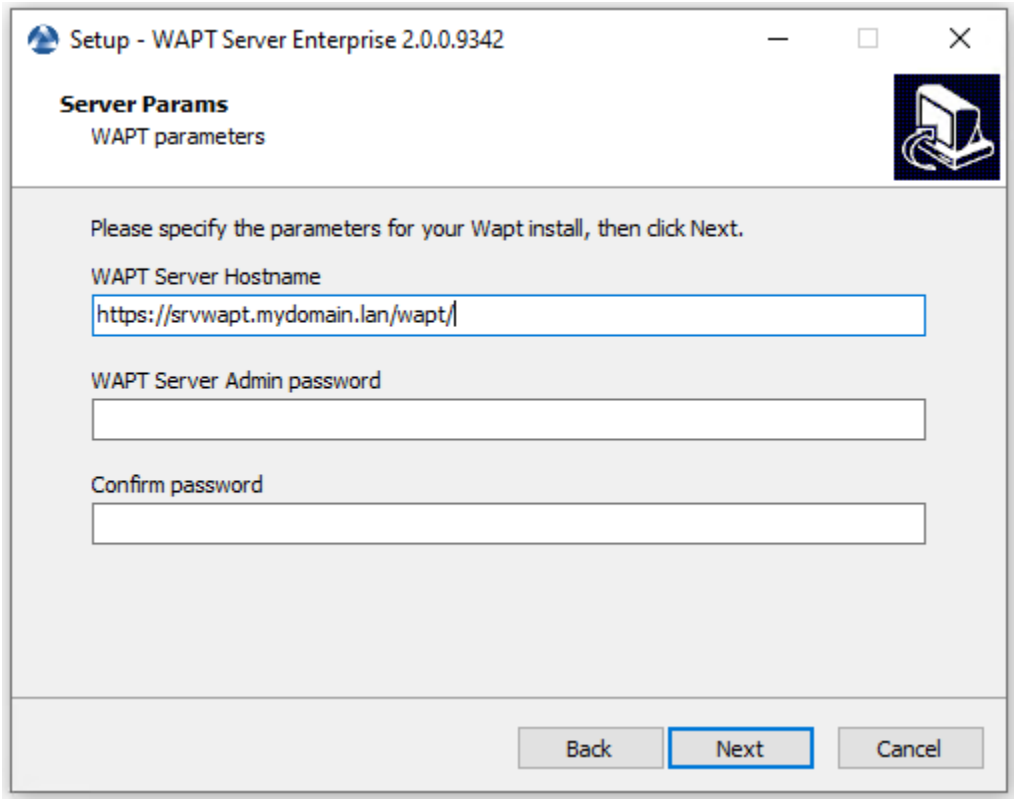


FIG. 7 – Modification du mot de passe du serveur WAPT

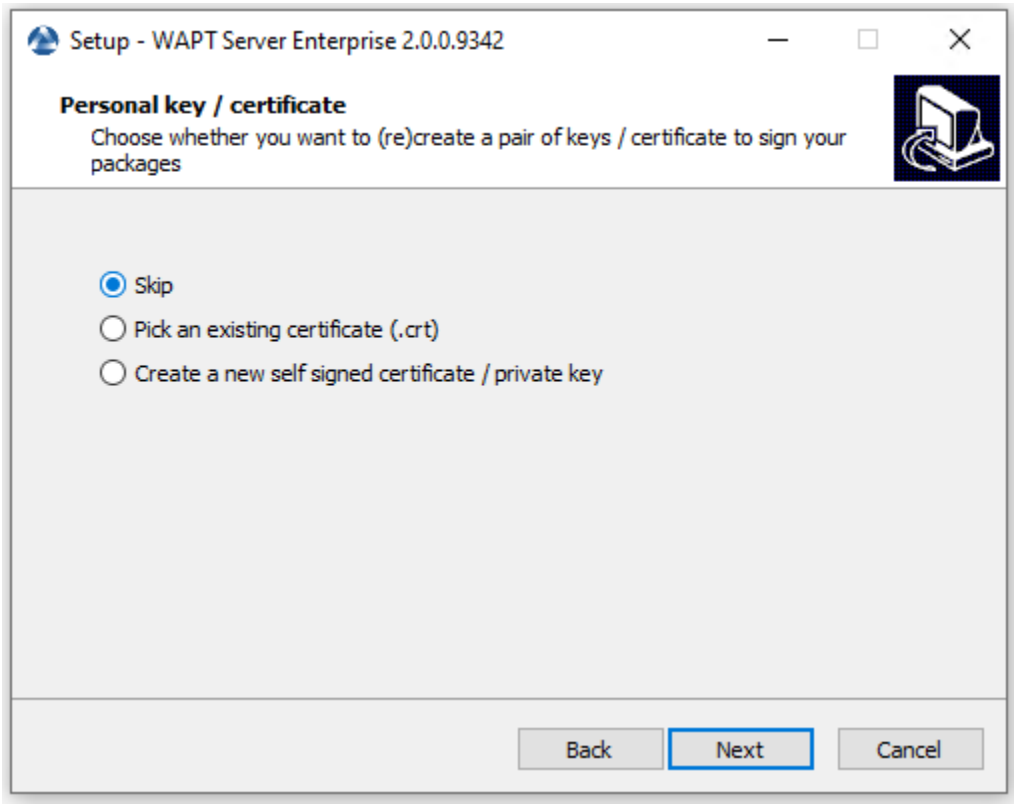


FIG. 8 – Sauter la création de la clé de signature

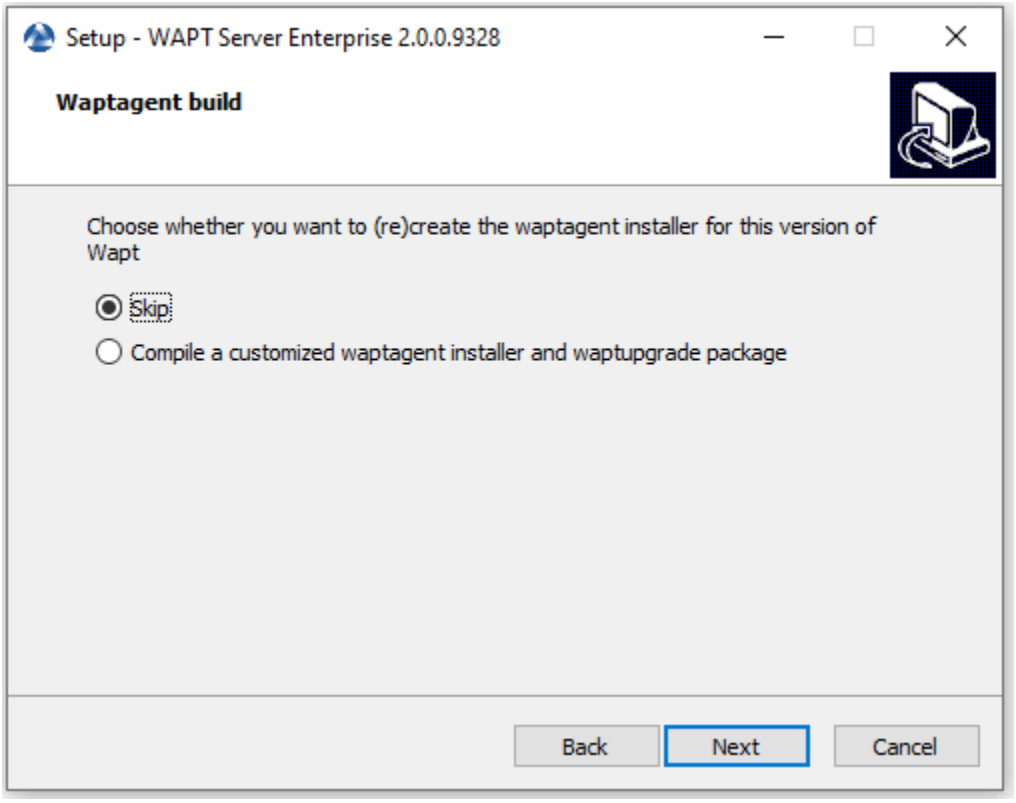


FIG. 9 – Sauter la construction de l'agent WAPT

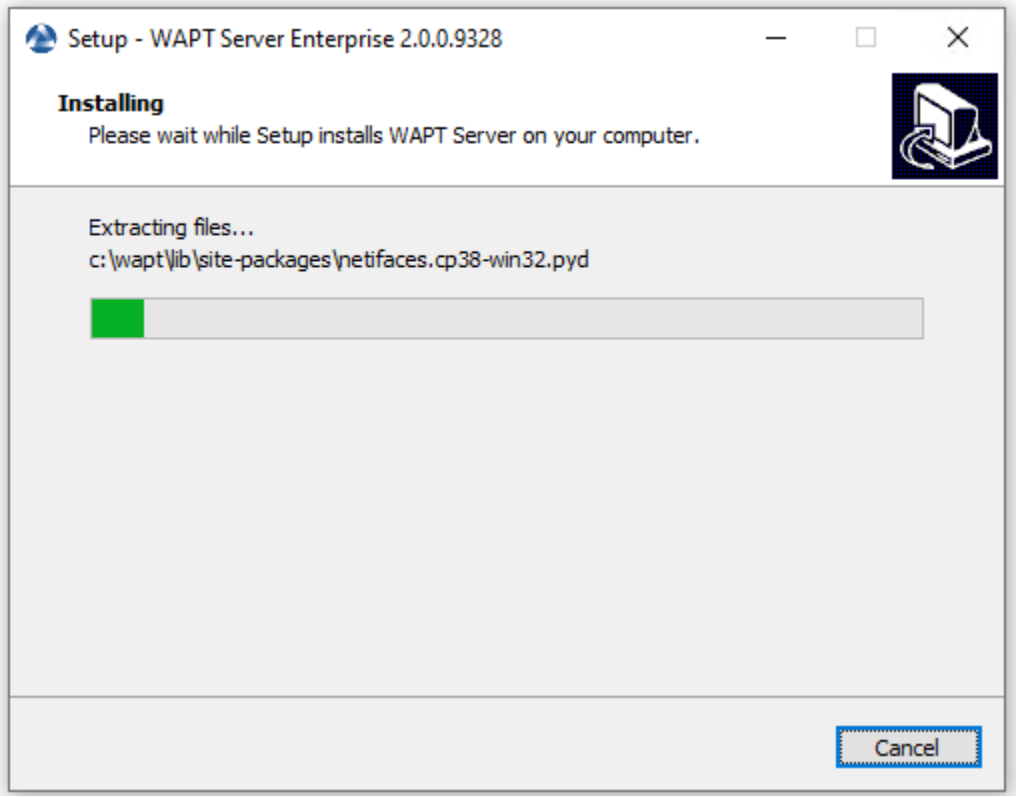


FIG. 10 – Installation du serveur WAPT en cours

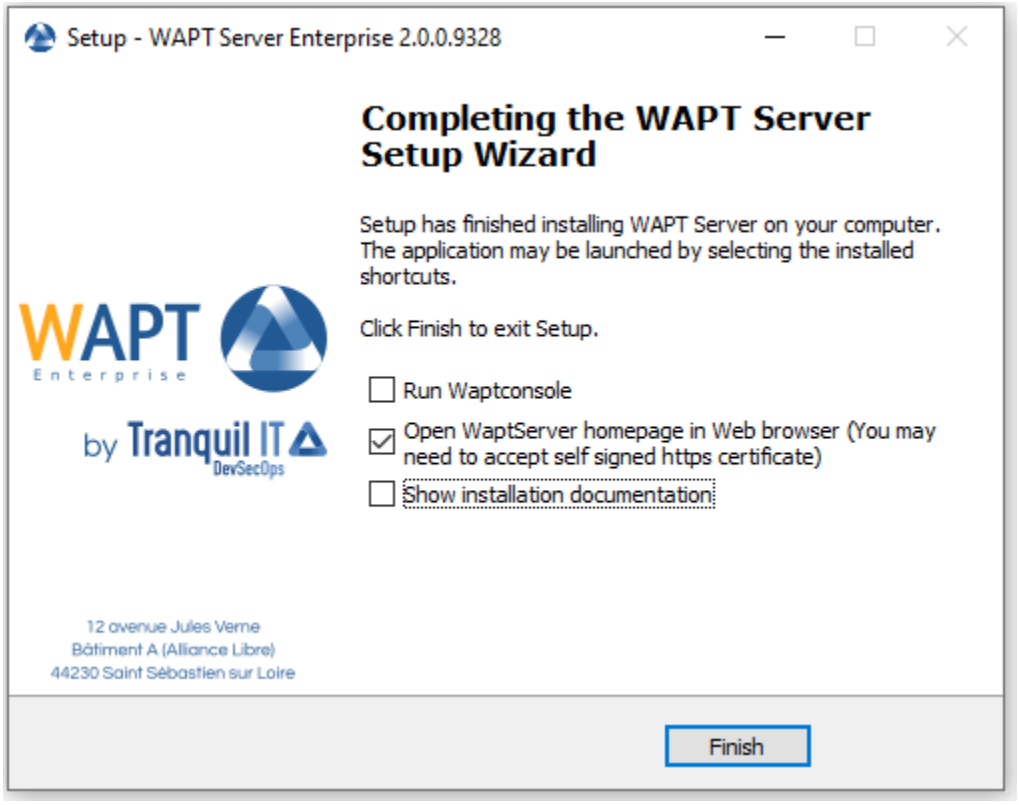
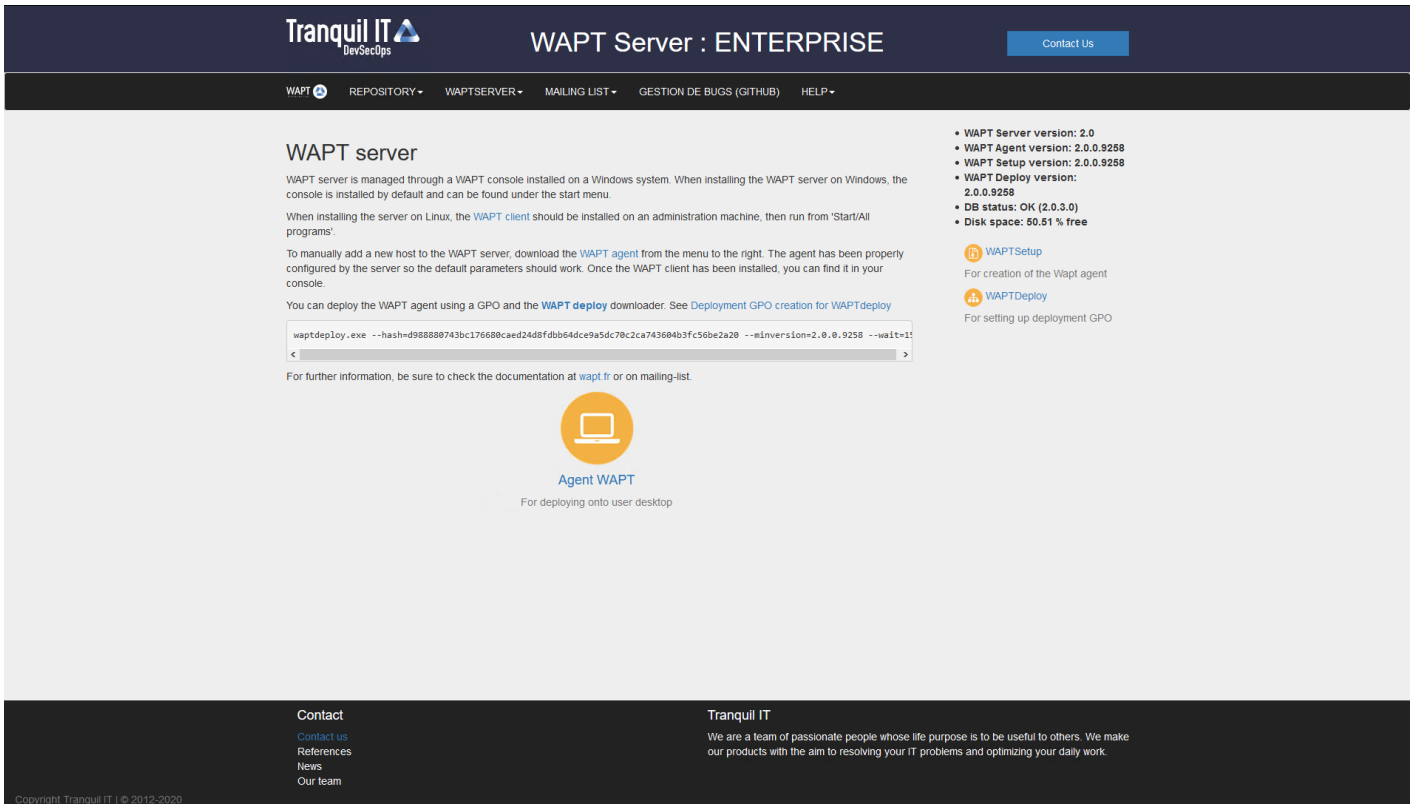


FIG. 11 – L'installation est terminée



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*

CHAPITRE 23

Mettre à jour de 1.6/1.7 vers 1.8

23.1 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.

23.1.1 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/entreprise/debian/wapt-1.8/ $(lsb_release␣
↪-c -s) main"  > /etc/apt/sources.list.d/wapt.list
apt update
apt install tis-waptserver tis-waptsetup
```

23.1.2 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
```

23.1.3 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 préfix 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.

23.2 CentOS

- Tout d'abord, mettez à jour la distribution sous-jacente CentOS/ RedHat.

```
yum update
```

23.2.1 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
```

23.2.2 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
```

23.2.3 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 préfix 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**.
— Using a **waptupgrade** package and deploying it using 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**.

23.3 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 : The WAPT Server may be installed on 64bit Windows 10 clients, and 64 bit Windows Server 2012/R2, 2016 and 2019 and 2022.

— 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-load :<https ://wapt.tranquil.it/wapt/releases/latest/waptserversetup.exe>* et lancez-le en tant que *Administrateur local*.
— Install the update.

Note : The procedure to follow is the same as the one for *installing a WAPT Server on Windows*.

Attention : Le préfix ne doit pas changer et vous NE DEVEZ PAS regénérer 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 préfix 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**.

CHAPITRE 24

Upgrading WAPT from versions older than 1.6

We recommend that you restart from scratch, that you provision an up-to-date WAPT Server environment alongside the old version, and that you use the old setup or a domain GPO to deploy the new WAPT agent.

CHAPITRE 25

Sauvegarder le serveur WAPT

Pour sauvegarder votre serveur, suivez cette procédure. Des sauvegardes régulières sont recommandées.

25.1 Linux

— Stop WAPT related services on the WAPT Server.

```
systemctl stop wapttasks
systemctl stop waptserver
systemctl stop nginx
```

— **Backup these directories using a backup tool**
(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/
```

— **Backup the PostgreSQL database using the `pg_dumpall` utility**
(adapt filename with your requirements).

```
sudo -u postgres pg_dumpall > /tmp/backup_wapt.sql
```

— Restart WAPT related services on the WAPT Server.

```
systemctl start wapttasks
systemctl start waptserver
systemctl start nginx
```

25.2 Windows

— Stop WAPT related services on the WAPT Server.

```
net stop wapttasks
net stop waptserver
net stop waptnginx
```

— Backup the WAPT repository folder on a remote backup destination.

```
C:\wapt\conf
C:\wapt\waptserver\repository
C:\wapt\waptserver\nginx\ssl
```

— Backup PostgreSQL Database with **pg_dump.exe**.

```
"C:\wapt\waptserver\pgsql-9.6\bin\pg_dumpall.exe" -U postgres -f C:\backup_wapt.sql
```

— Restart WAPT related services on the WAPT Server.

```
net start wapttasks
net start waptserver
net start waptnginx
```

CHAPITRE 26

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.

26.1 Linux

— Stop WAPT related services on the WAPT Server.

```
systemctl stop nginx
systemctl stop waptserver
systemctl stop wapttasks
```

— Restore the following directories.

```
    # 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/
```

— Restore the database (adapt the name of your file). The first command **deletes** the WAPT database (if it exists). Make sure that your dump file is correct before deleting !

```
sudo -u postgres psql -c "drop database wapt"
sudo -u postgres psql < /tmp/backup_wapt.sql
```

— Apply ownership rights to the restored folders.

```
    # 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/
```

— Scan package repositories.

```
    # Debian / Ubuntu
wapt-scanpackages /var/www/wapt/

    # CentOS / RedHat
    wapt-scanpackages /var/www/html/wapt/
```

— Restart WAPT related services on the WAPT Server.

```
systemctl start wapttasks
systemctl start waptserver
systemctl start nginx
```

26.2 Windows

— Stop WAPT related services on the WAPT Server.

```
net start wapttasks
net start waptserver
net start waptnginx
```

— Restore the following directories.

```
C:\wapt\waptserver\repository
C:\wapt\waptserver\conf
C:\wapt\waptserver\nginx\ssl
```

— Apply full rights to the folder `C:\wapt\waptserver\repository` for the « Network Service » group.
— Restore PostgreSQL Database with **`pg_restore.exe`**.

```
"C:\wapt\waptserver\pgsql-9.6\bin\psql.exe" -f c:\backup_wapt.sql -U postgres
```

— Scan package repositories.

```
wapt-scanpackages "C:\wapt\waptserver\repository\wapt"
```

— Restart WAPT related services on the server.

```
net start wapttasks
net start waptserver
net start waptnginx
```

CHAPITRE 27

Uninstalling the WAPT agent from clients

27.1 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`.

— Default silent uninstall of a WAPT agent can be achieved with the following command.

```
unins000.exe /VERYSILENT
```

— An additional argument can be passed to **unins000.exe** to cleanup everything.

```
unins000.exe /VERYSILENT /purge_wapt_dir=1
```

TABLEAU 1 – Liste complète des arguments de ligne de commande pour **unins000.exe**

Paramètres	Description
/VERYSILENT	Launches **unins000.exe** silently.
/purge_wapt_dir = **1**	Purges the WAPT directory (removes all folders and files).

— It is possible to use a package for this.

```
# -*- 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)
```

27.1.1 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()
```

27.2 Linux

— **Default uninstall of a WAPT agent can be achieved with the following command,**
depending on your Linux OS :

```
            # Debian / Ubuntu
apt remove --purge tis-waptagent

    # CentOS / Redhat
yum remove tis-waptagent
```

— An additional step can be done using these commands (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
```

27.3 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 paquet*.

CHAPITRE 28

Déployer l'agent WAPT

28.1 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'éxécutable 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.

28.1.1 Manuellement

Attention : Manually installing the WAPT agent requires *Local Administrator* rights on the computer. Manually installing the WAPT agent using a Domain Admin account **WILL NOT WORK**.

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.

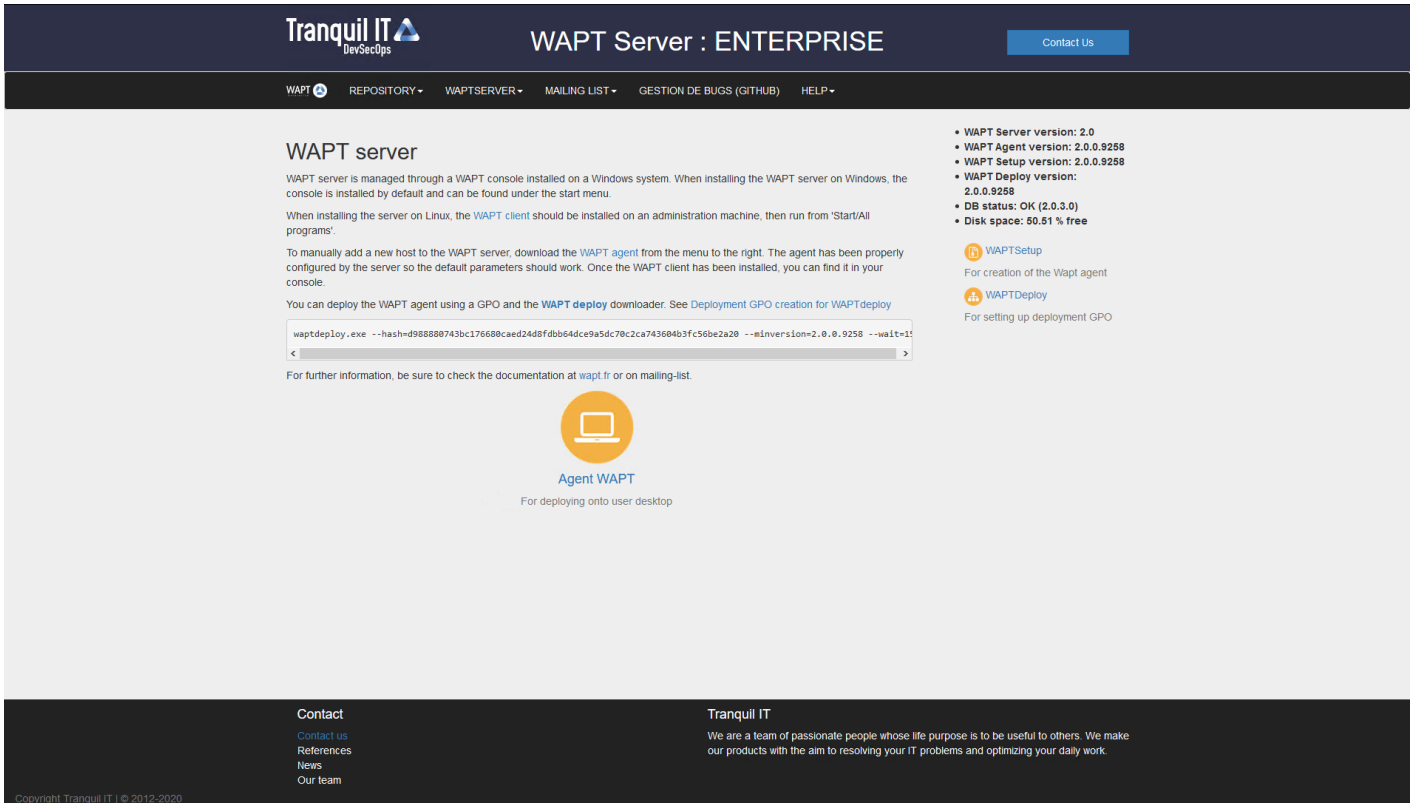


FIG. 1 – Télécharger l'agent WAPT à déployer sur les ordinateurs

— 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 *bug du BIOS avec l'UUID*	Non coché

— Choisissez le dépôt WAPT et le serveur WAPT et cliquez sur *Suivant* pour passer à l'étape suivante ;
— Installez l'agent WAPT en cliquant sur *Installer* ;
— Attendez que l'installation de l'agent WAPT se termine, puis cliquez sur *Terminer* pour quitter ;

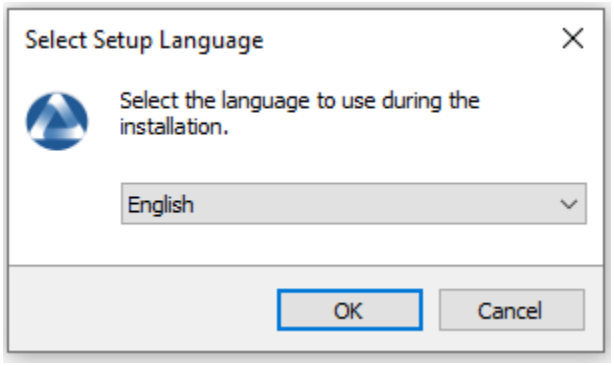


FIG. 2 – Choisissez la langue d'installation

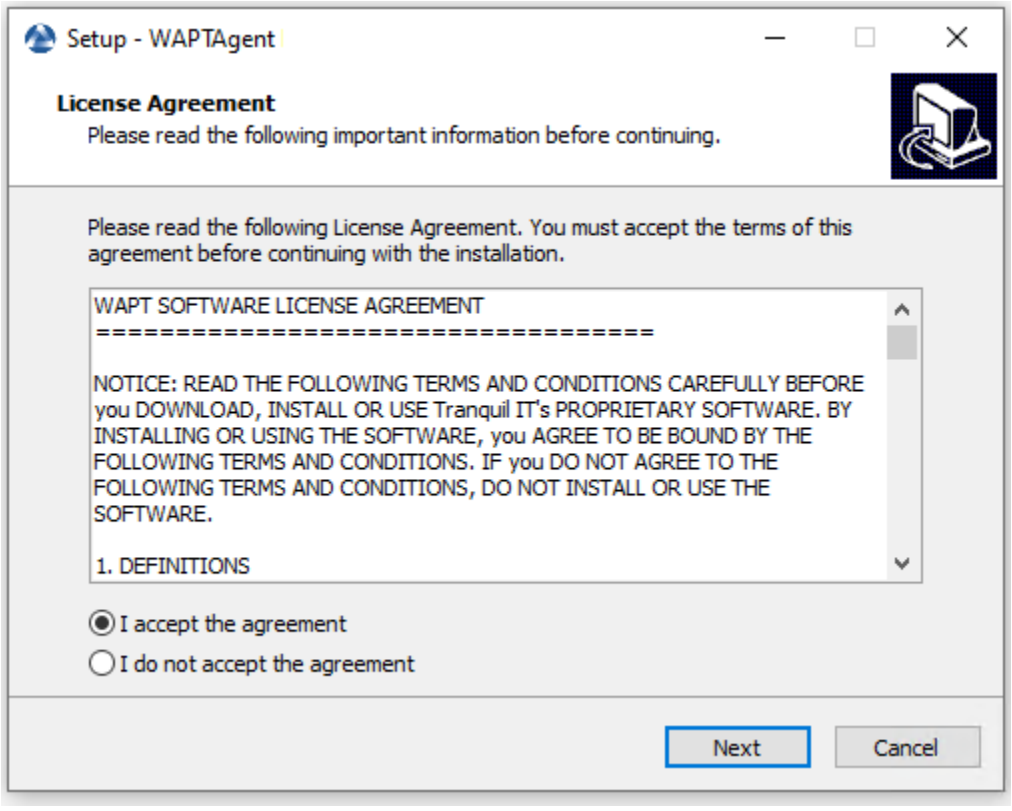


FIG. 3 – Accepter le CLUF

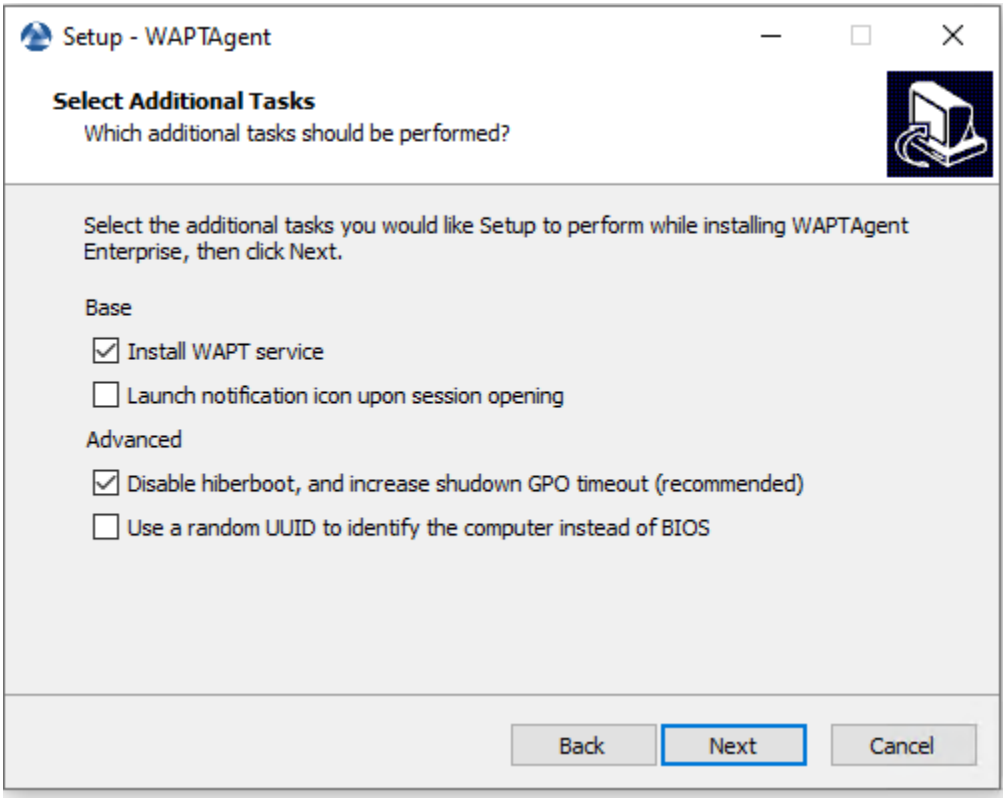


FIG. 4 – Choisissez les options de l'installateur

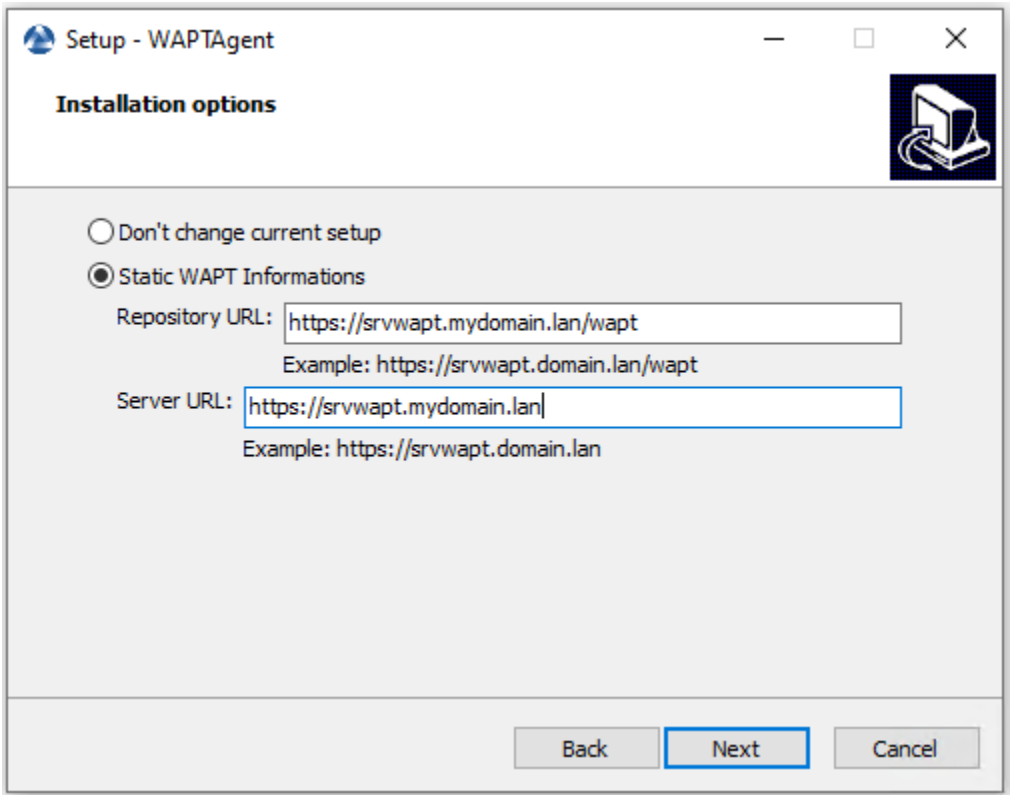


FIG. 5 – Choisir le dépôt et le serveur WAPT

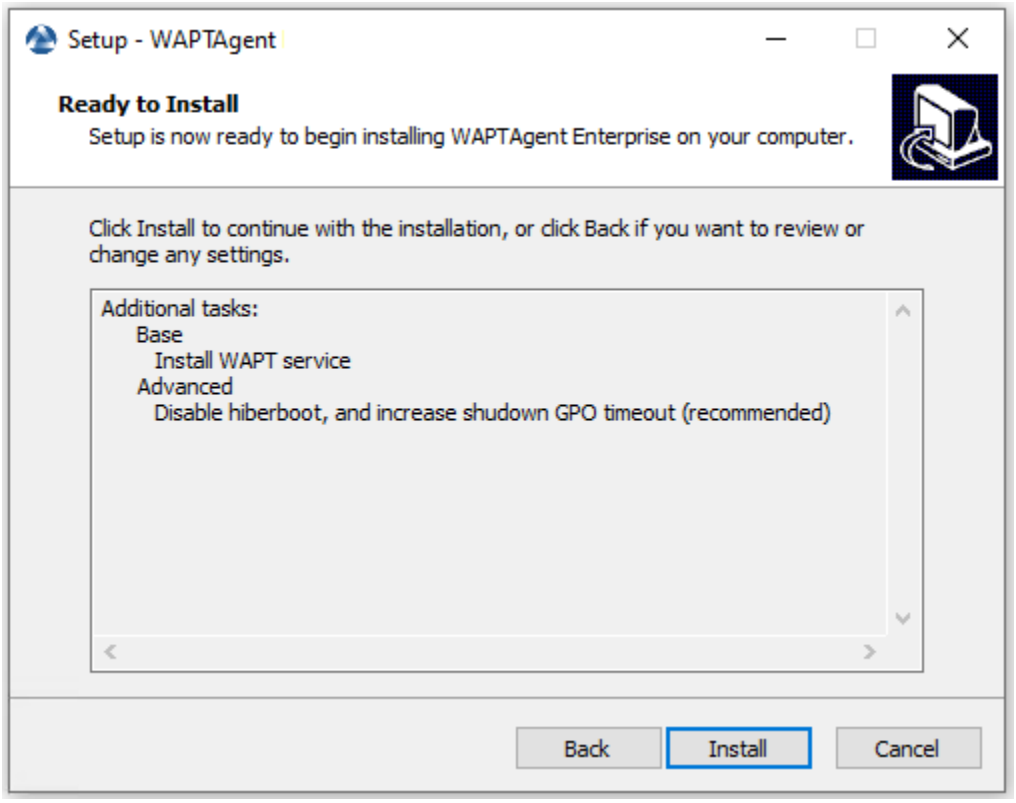


FIG. 6 – Résumé des options d'installation

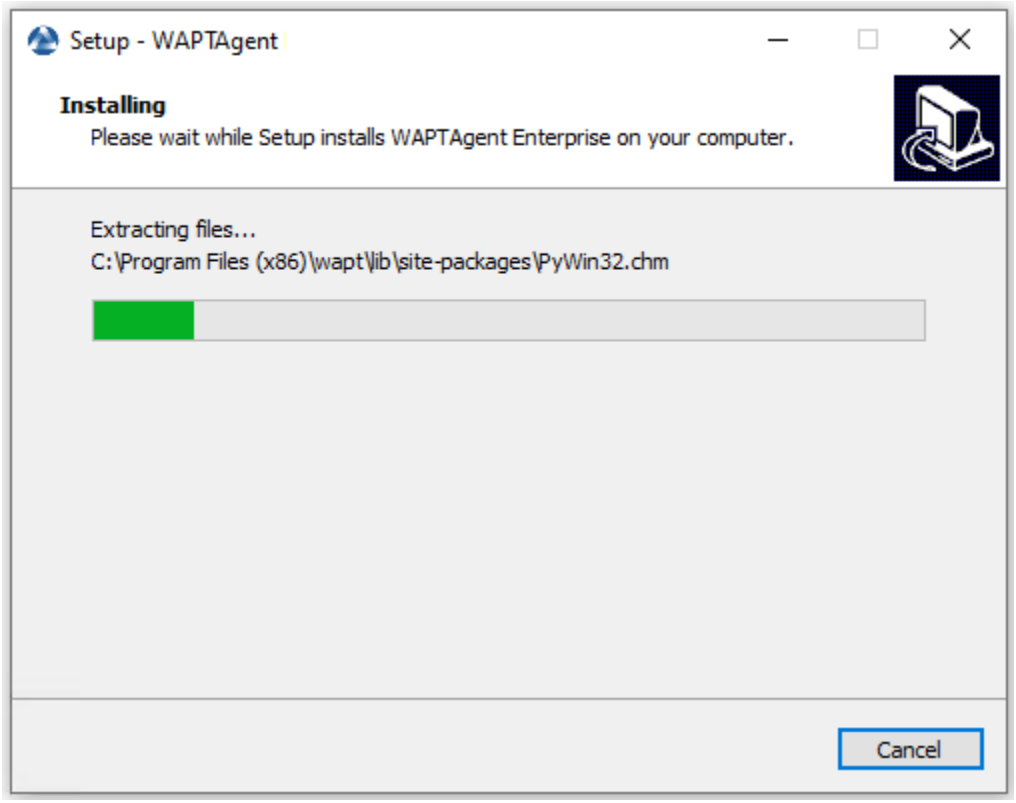


FIG. 7 – Installation en cours

L'installation de l'agent WAPT est terminée. L'enregistrement de la machine auprès du serveur WAPT se fait automatiquement.

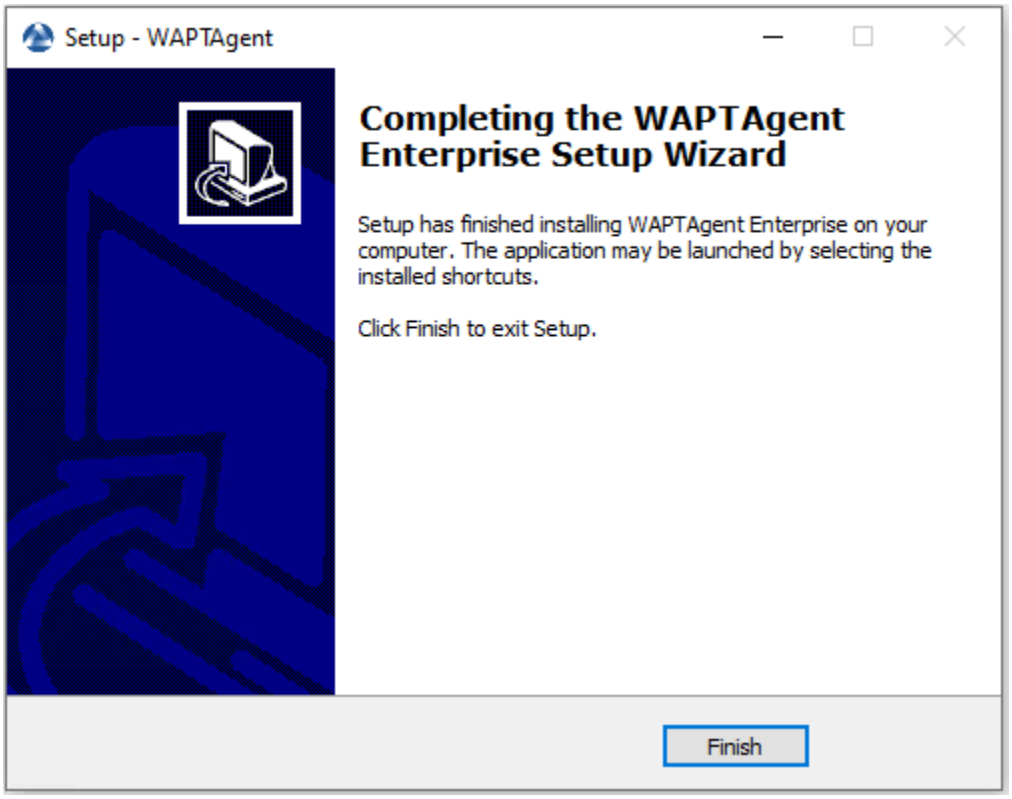


FIG. 8 – Fin de l'installation de l'agent WAPT

Pour gérer les clients WAPT de votre organisation, consultez la *documentation sur l'utilisation de la console WAPT*.

28.1.2 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* ;

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 remplit lors de l'installation.
/wapt_server = https://srvwapt.mydomain.lan	**URL du serveur WAPT dans wapt-get.ini rempli pendant l'installation** domaine dans wapt-get.ini rempli lors de 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 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**.

Pour en savoir plus sur les options disponibles avec *InnoSetup*, visitez cette documentation

Avec waptdeploy

waptdeploy est un petit binaire qui :

— Vérifie la version de l'agent WAPT.
— Télécharge via https l'installeur **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 conseillons d'utiliser une GPO.

Télécharger `waptdeploy.exe` depuis votre page d'accueil du serveur WAPT.

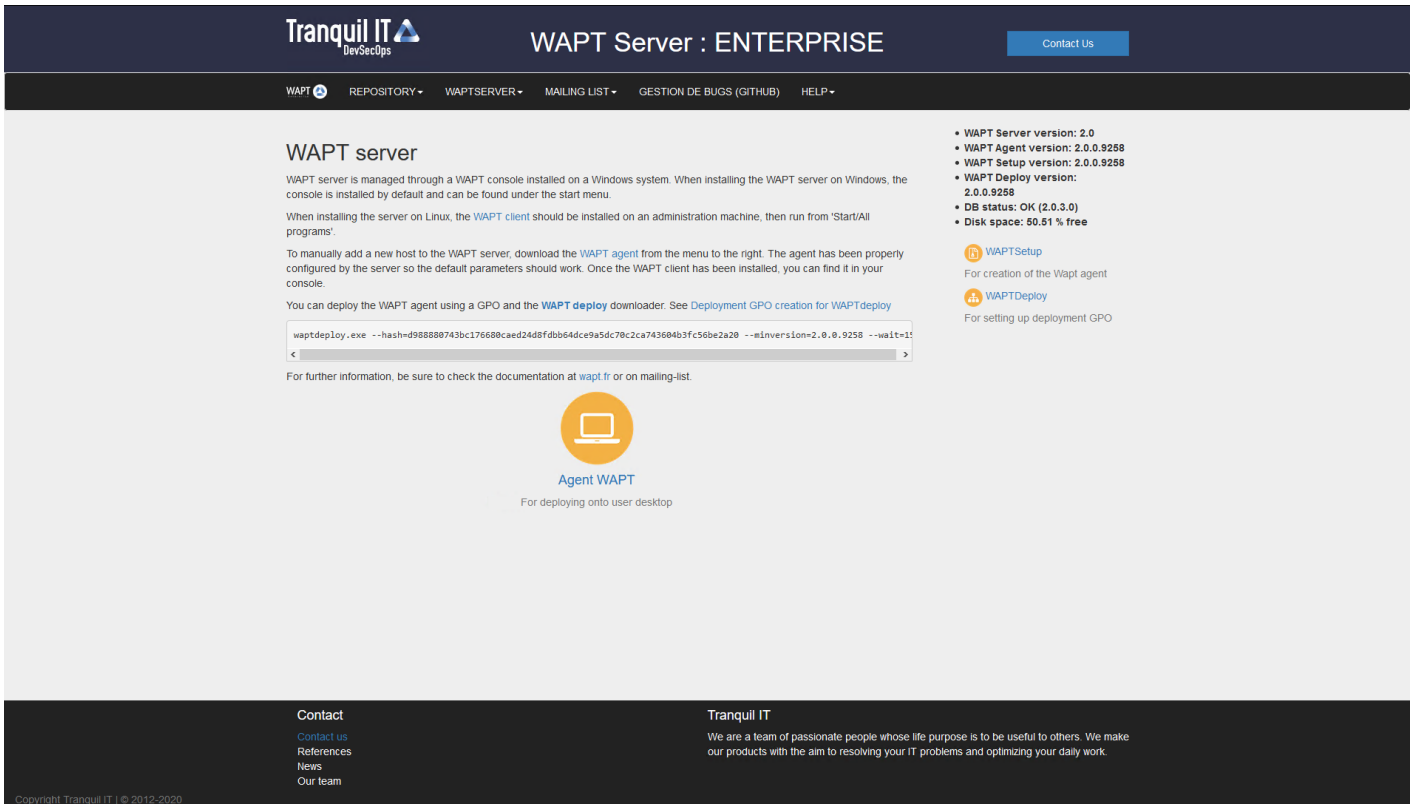


FIG. 9 – Télécharger waptdeploy

Avec une GPO

— **Créer une nouvelle stratégie de groupe sur le serveur Active Directory (Microsoft** Active Directory or 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`.

Fig. 13 – Sélection du script waptdeploy.exe

— Cliquez sur *Ouvrir* pour confirmer l'importation du binaire **`waptdeploy`** .

Indication :

Il est nécessaire de renseigner la somme de controle du `waptagent.exe`
en tant qu'argument de la GPO *waptdeploy*.

Cela va empêcher l'hôte distant d'exécuter un fichier erronné / corrompu `waptdeploy`.

```
--hash=checksum WaptAgent --minversion=1.2.3 --wait=15 --waptsetupurl=http://srvwapt.mydomain.lan/
↪wapt/waptagent.exe
```

Paramètres et somme de contrôle du `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.

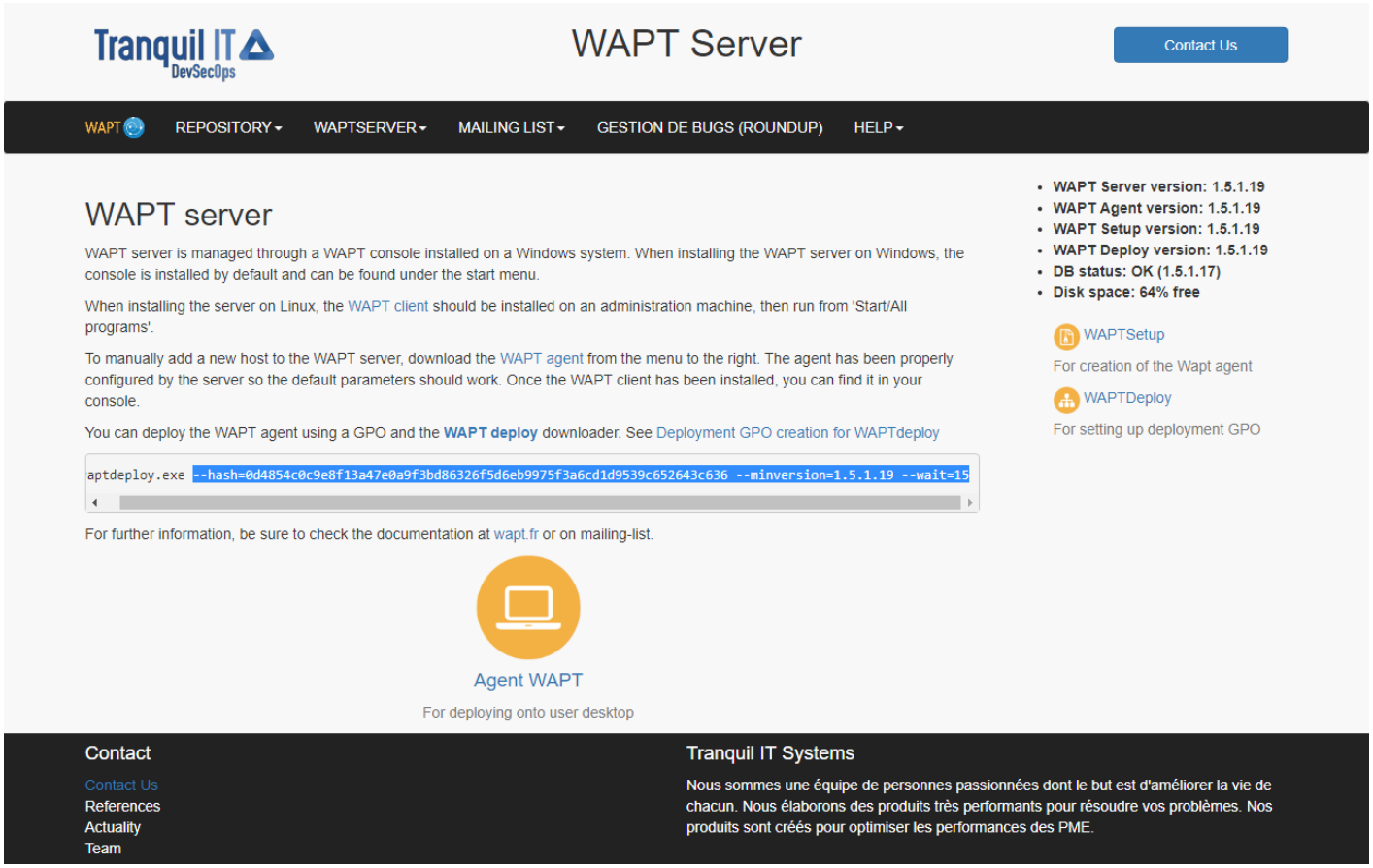


FIG. 14 – Console web du serveur WAPT

— Copier les paramètres requis.

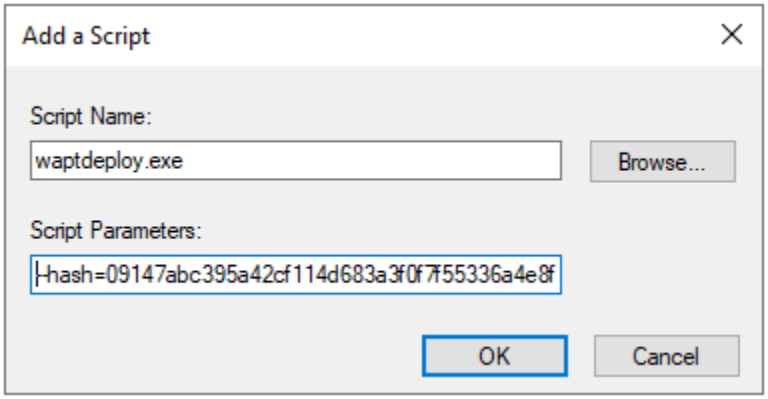


FIG. 15 – ajouter le script *waptdeploy* à la GPO de démarrage

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

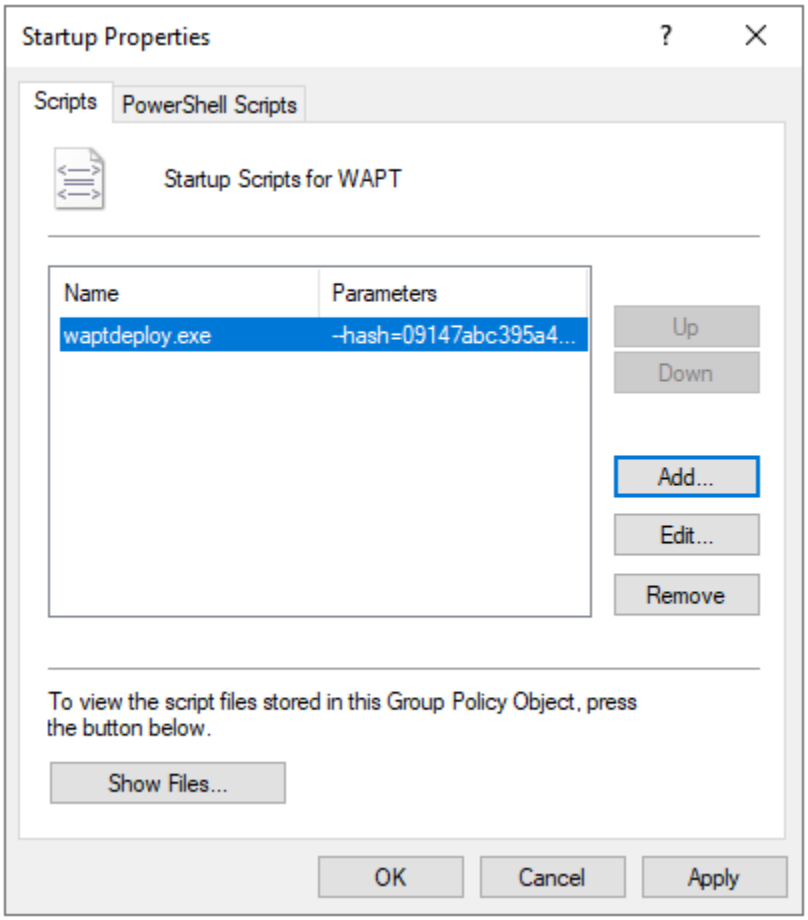


FIG. 16 – La GPO WAPTdeploy sera déployée au prochain démarrage

— 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
et d'arrêt sur la GPO.

Indication : Plus d'arguments sont disponible pour waptdeploy

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` = autorunTray,installService,installredist2008,autoUpgradePolicy	**S'il est donné, passez ces arguments aux options /TASKS de** l'installateur de waptagent. Défaut = installService,installredist2008,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= :file :*<repo_url>/waptagent.exe*.

```
--hash="43254648348435423486"--minversion=2.0 --waptsetupurl=http://srvwapt.mydomain.lan/wapt/
↪waptagent.exe --wait=10
```

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.

— Copy `waptdeploy.exe` and `waptagent.exe` in the netlogon share of your Active Directory Server (`\mydomain.lan\netlogon\waptagent.exe`).
— **Créer une nouvelle stratégie de groupe sur le serveur Active Directory (Microsoft** Active Directory or Samba-AD).
— Add a new strategy with *Computer configuration* → *Preferences* → *Windows Settings* → *Files*.
— Créer un nouveau fichier pour copier waptdeploy.

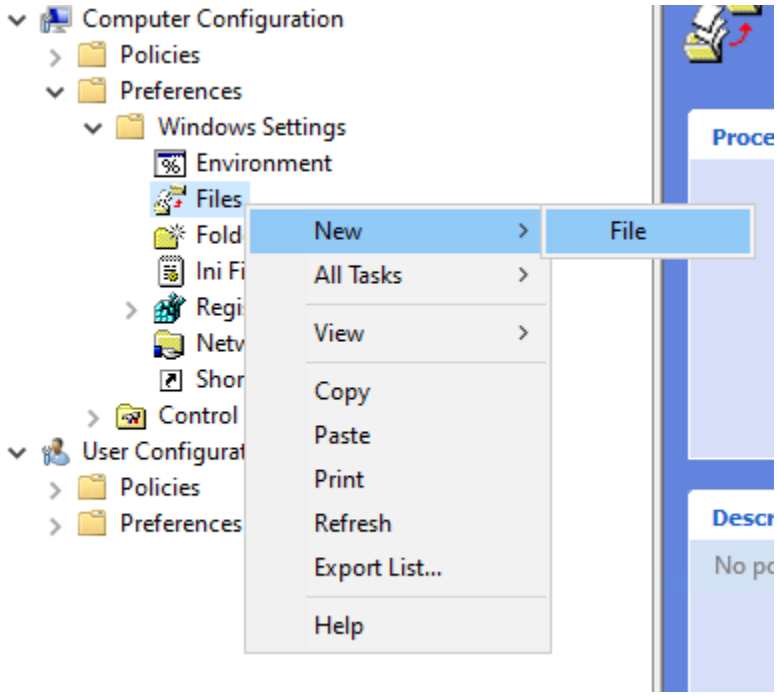


FIG. 17 – Nouveau fichier dans la GPO

— 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

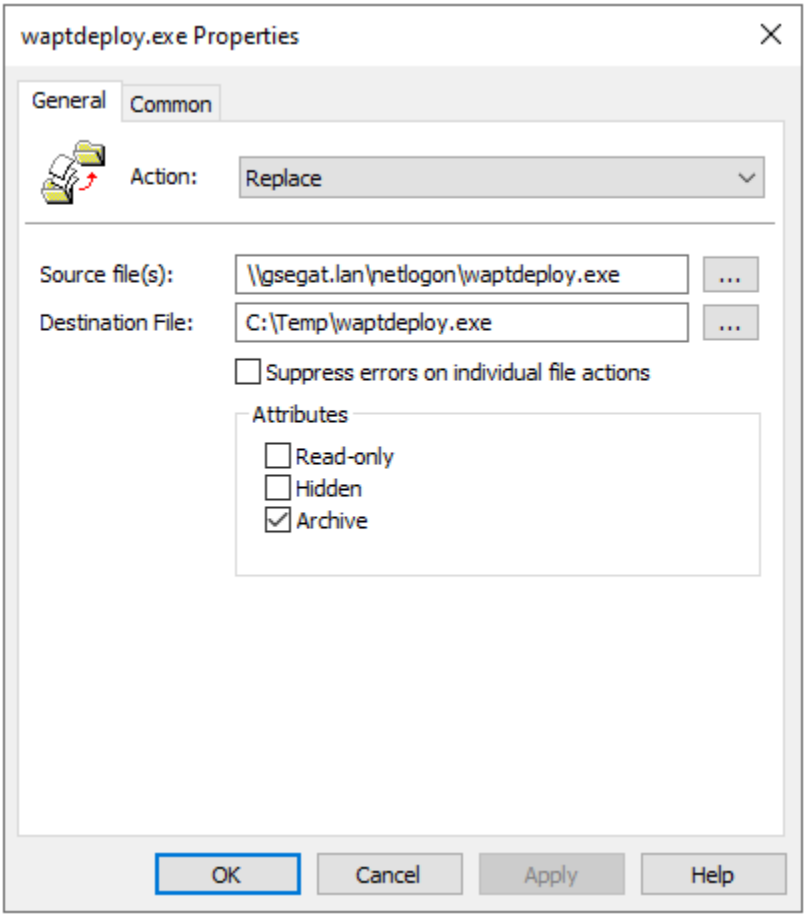


Fig. 18 – Progression de l'installation de l'agent WAPT

— Créer un nouveau fichier pour copier waptagent.

— Définir les paramètres.

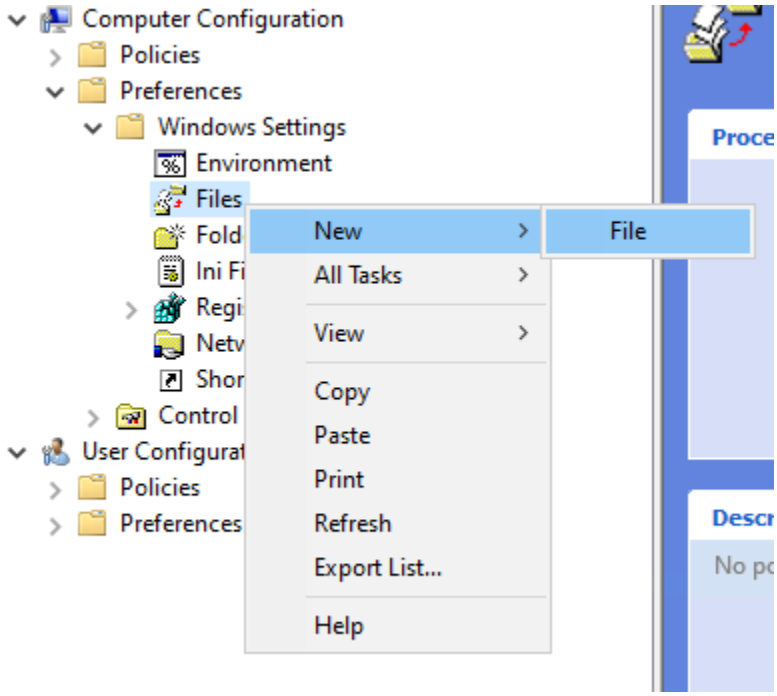


FIG. 19 – Nouveau fichier dans la GPO

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

— Then go to the Scheduled Task menu with *Computer configuration → Preferences → Control Panel Settings → Scheduled Tasks*.
— Create a new Scheduled Task with *Right-click → New → Scheduled Task (At least Windows 7)*.

— Définir `Action` sur `Remplacer`.
— **Pour *Lors de l'exécution de la tâche, utilisez le compte d'utilisateur suivant***
collez utiliser *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 allez dans l'onglet *Déclencheurs*.
— 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*, selectionnez *1 jour*.
— **Cocher *Arrêter la tâche si elle s'exécute plus de***

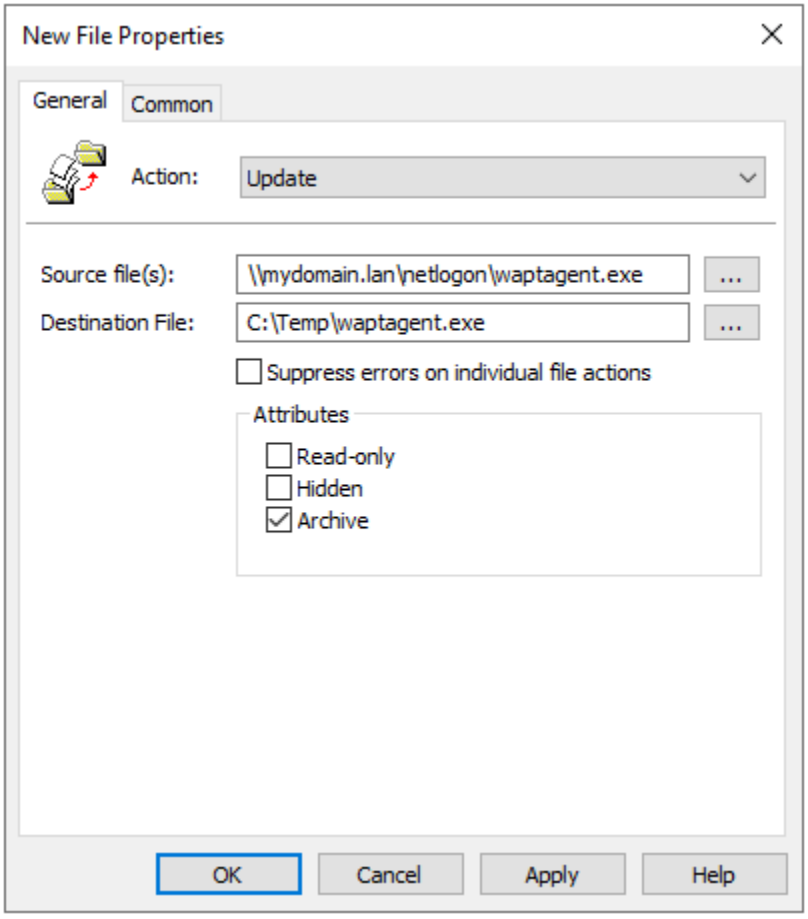


FIG. 20 – Progression de l'installation de l'agent WAPT

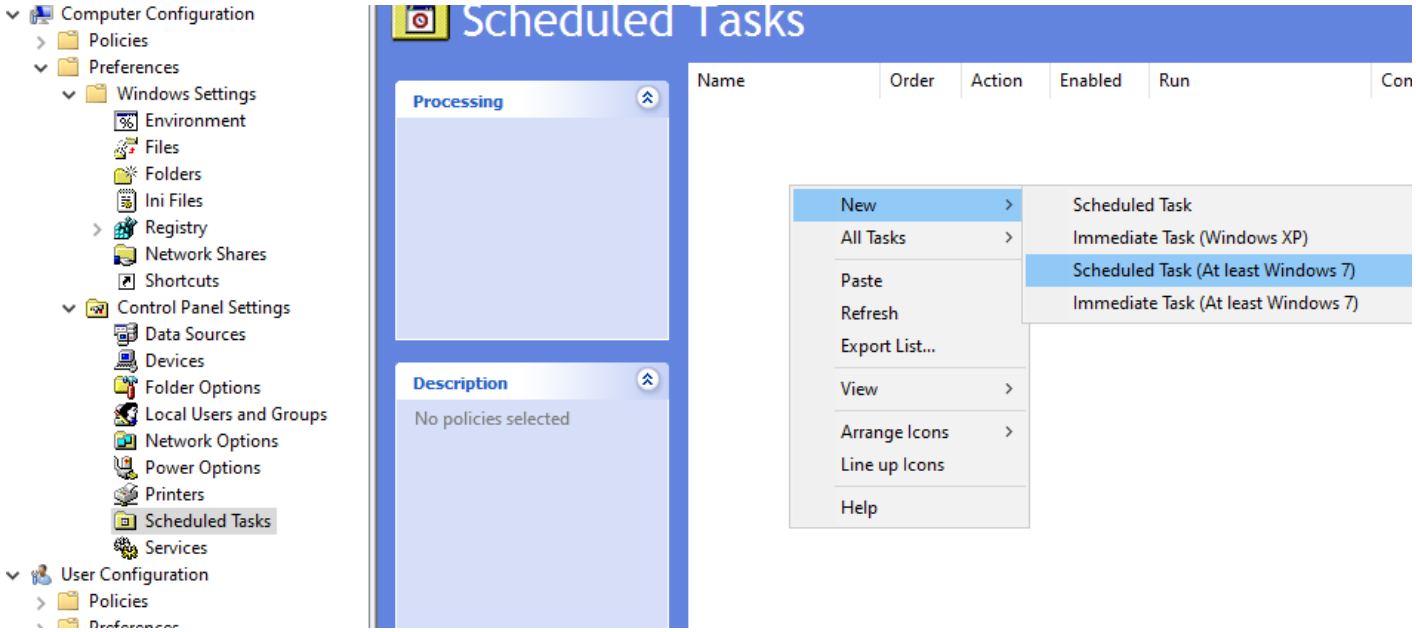


FIG. 21 – Tâche Créer dans la fenêtre des propriétés de *deploywapt*

et sélectionner *2 heures*.

— **Vérifier que *Enabled* est coché,**
puis aller dans l'onglet *Actions*.

— Créer une nouvelle action *Démarrer un programme* pour :file`waptdeploy.exe`.

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 controle 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 l'installeur `waptagent.exe` à utiliser
pour la GPO *waptdeploy* sont disponibles sur le serveur WAPT en visitant https://srvwapt.mydomain.lan.

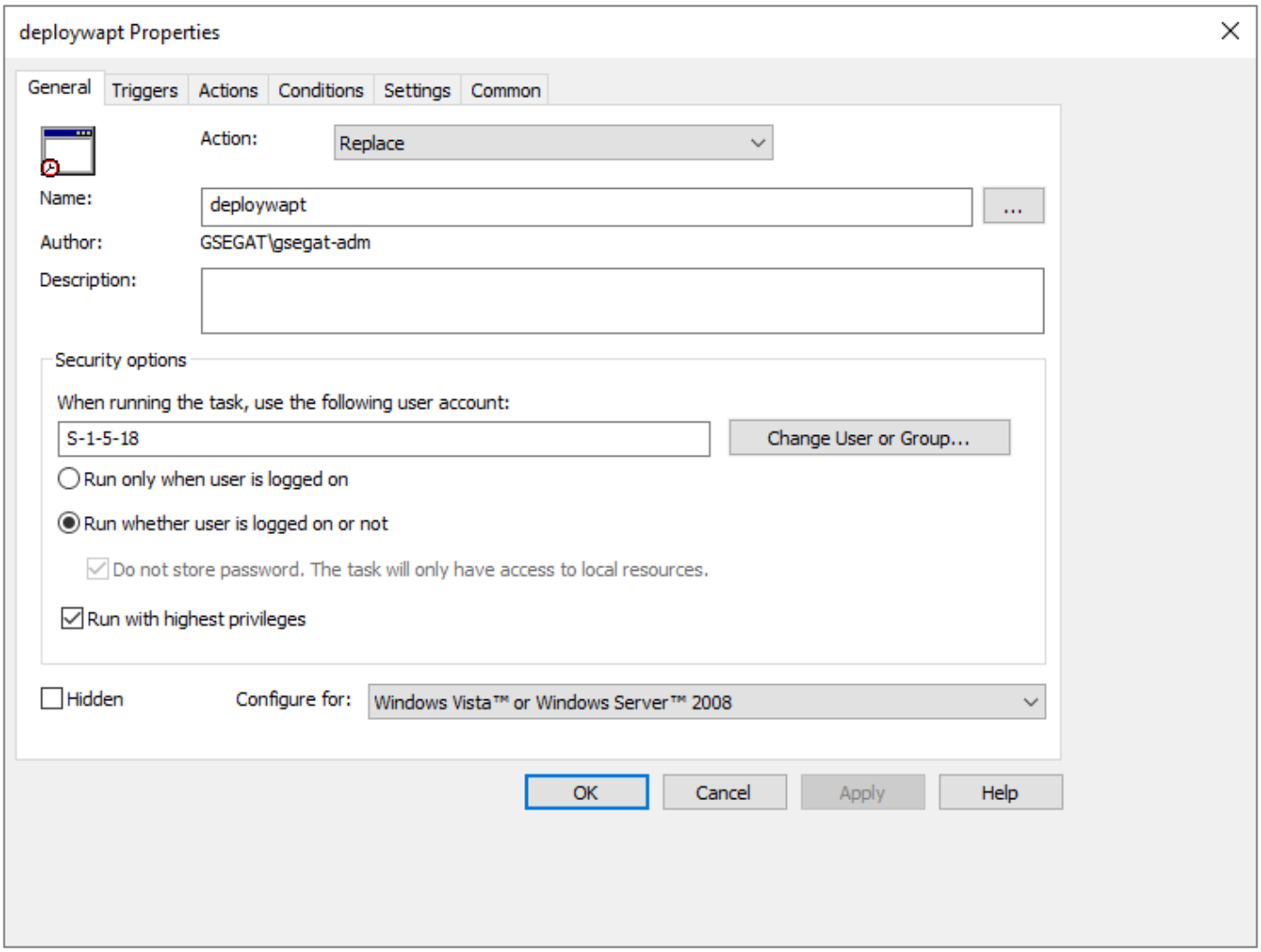


FIG. 22 – Onglet Général de la fenêtre de propriété de *deploywapt*

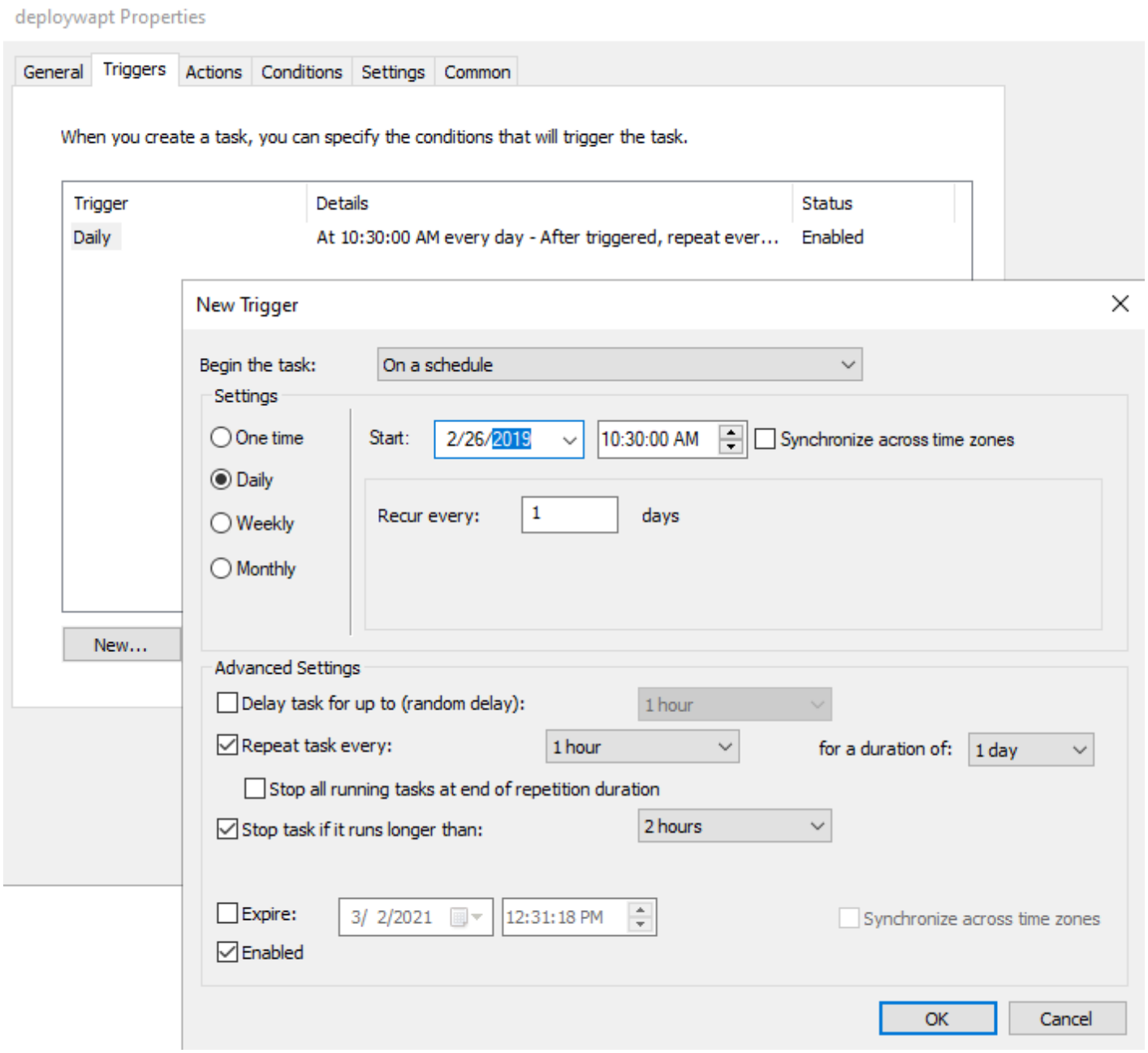
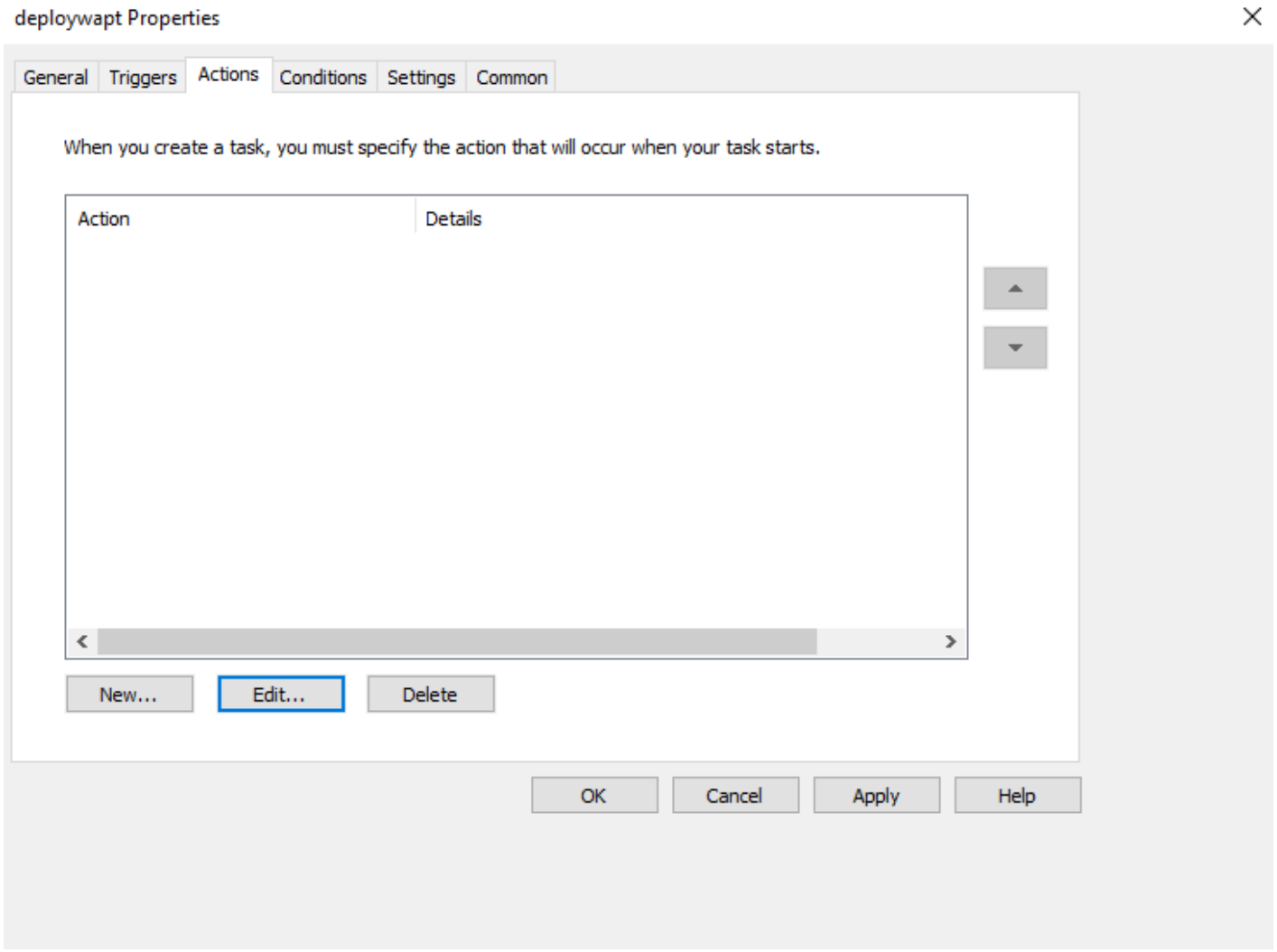


Fig. 23 – Onglet Déclencheur de la fenêtre de propriété *deploywapt*


deploywapt Properties
General
Triggers
Actions
Conditions
Settings
Common
When you create a task, you must specify the action that will occur when your task starts.
Action
Details
New...
Edit...
Delete
OK
Cancel
Apply
Help

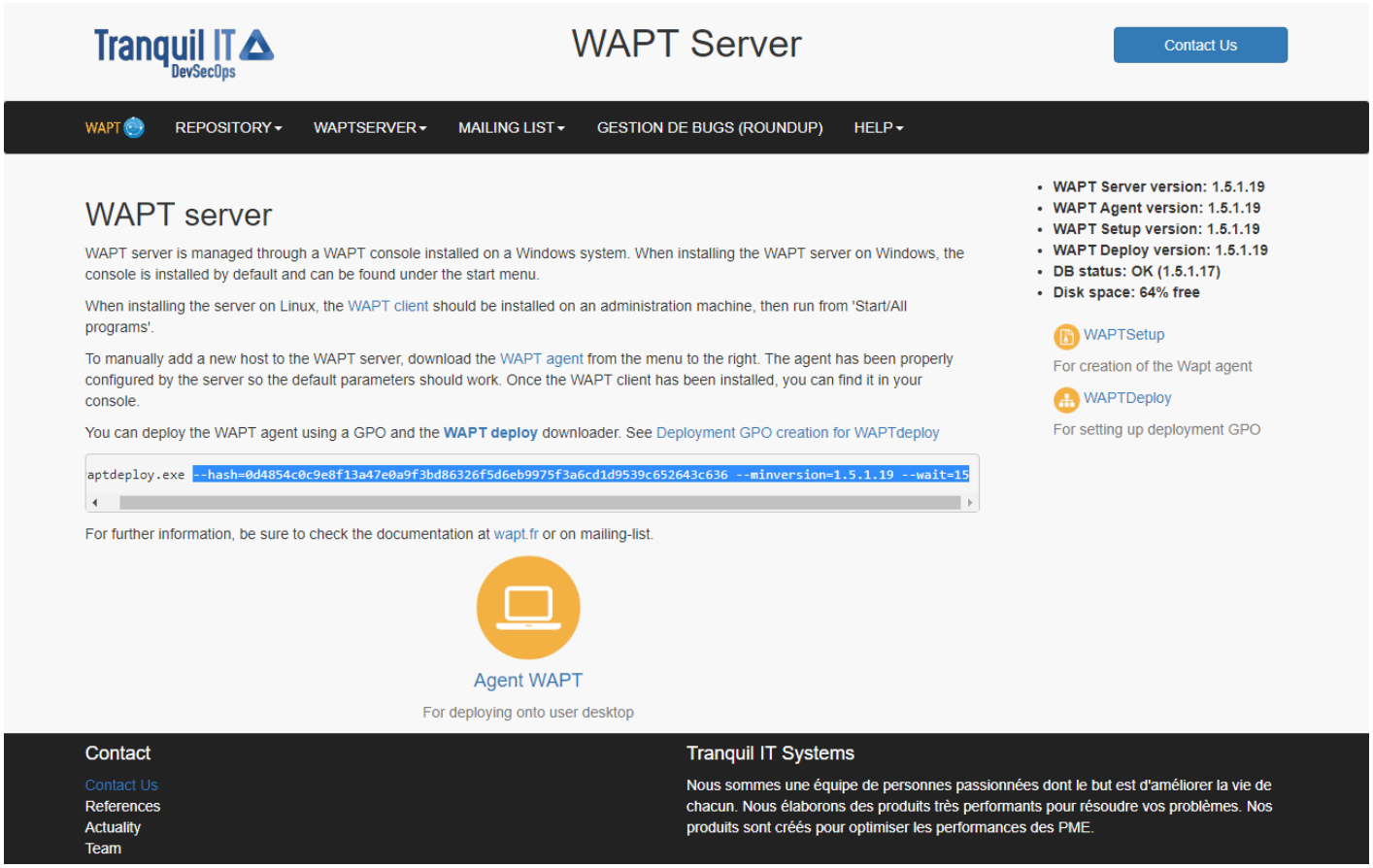


FIG. 24 – Console web du serveur WAPT

— **Copiez les paramètres requis et changez `waptsetupurl`**
en `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` = autorunTray,installService,installredist2008,autoUpgradePolicy	**S'il est donné, passe cet argument aux options /TASKS** of the waptagent installer. Default = installService,installredist2008, 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 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`).

— Aller à l'onglet *Paramètres*.

— In the *Settings* tab, only check *Run task as soon as possible after a scheduled start is missed*.

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 **`Task Schedeler`** en mode administrateur.

28.2 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 ;

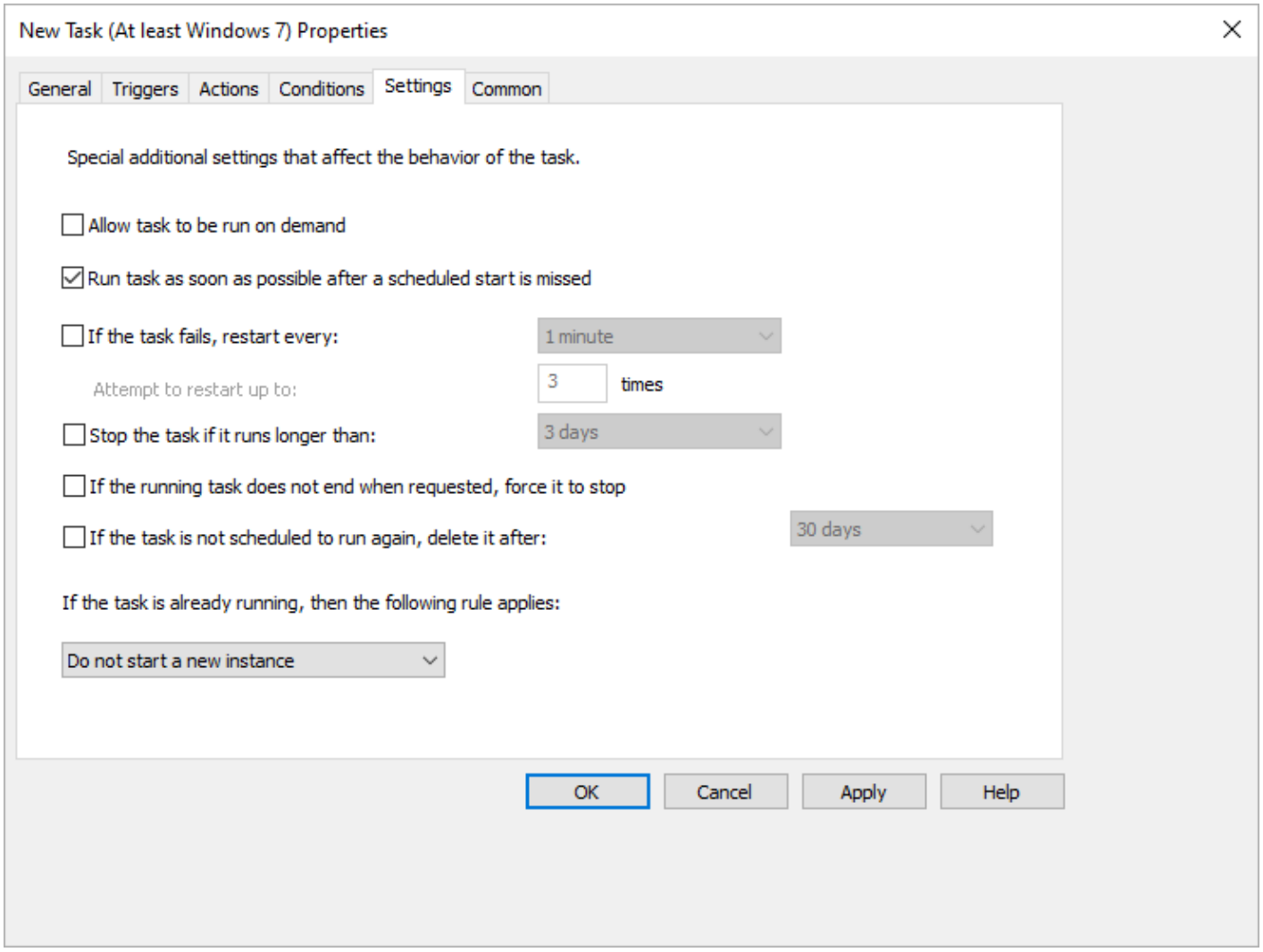


FIG. 25 – Onglet Paramètres de la fenêtre de propriété *deploywapt*

28.2.1 Debian

Discovery

Important :

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

— 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
```

— 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
```

— 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 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.

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
```

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.

28.2.2 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
```

— 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
```

— 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 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.

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
```

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.

28.2.3 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-2.0/
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 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.

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
```

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.

28.3 MacOS

Attention : Pour l'instant, l'agent n'a été testé que sur :

— High Sierra (version 10.13) ;
— Mojave (10.14) ;
— Catalina (10.15) ;
— Big Sur (version 10.16).

28.3.1 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/

28.3.2 Enterprise

Important : A la date du 2024-01-09 vous devez utiliser le dernier dépôt nitghtly 2.1 :

```
https://srvwapt-pro.tranquil.it/entreprise/nightly/
```

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-enterprise-version-macos-hash.pkg` ;
— créer le lien complet avec :

1. `https://srvwapt-pro.tranquil.it/entreprise/release/latest/`

et

2. `tis-waptagent-enterprise-version-macos-hash.pkg` copied previously ;

— 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`

28.3.3 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.

28.3.4 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 ;

28.3.5 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.

CHAPITRE 29

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.

29.1 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.

29.2 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 dabord, le paquet copie le nouveau fichier `waptagent.exe` sur le poste et créé 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.

CHAPITRE 30

Utiliser la console WAPT

To install and start the WAPT console visit the documentation for *installing the WAPT console*.

Note : Si vous avez passé l'étape de la création de l'agent WAPT, retournez à la documentation sur *la construction de l'installeur 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`**.

30.1 Ajouter un paquet à la machine

If you want to add WAPT packages directly on the host, you have to edit the host package.

To do so, you have 3 methods :
- Double-click on the host.
- Right-click on the host then *Edit host*.
- Select a host and use the *Edit host* button.

Then, you just have to drag and drop wanted package(s) and confirm.

Pressing *Save* does the same thing as doing an *update*.

Pressing *Save and apply* does the same thing as an *update* immediatelly followed by an *upgrade*.

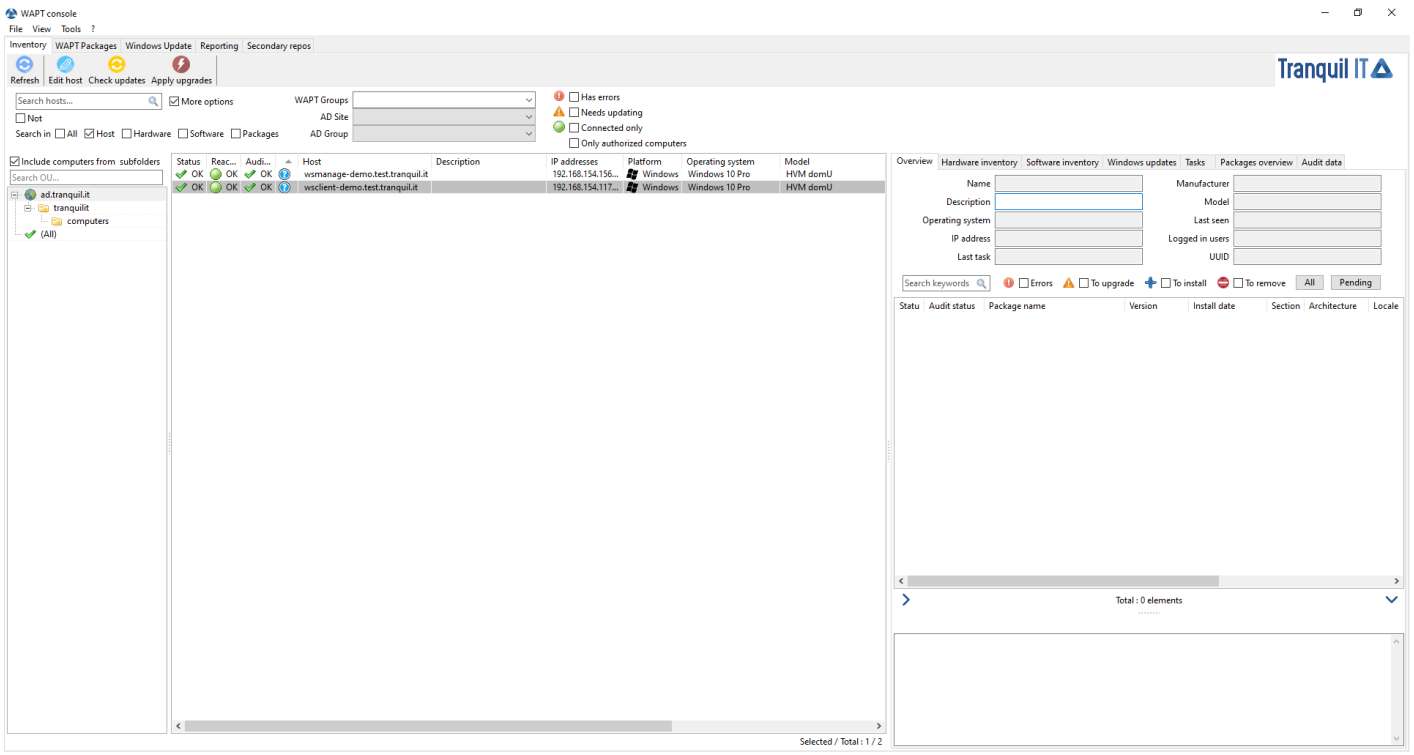


Fig. 1 – Inventaire des hôtes enregistrés avec WAPT

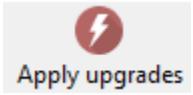
30.2 Vérifier les mises a jour sur l'hote

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*.

30.3 Appliquer les maj sur l'hote



Ce bouton exécute les mises à jour en attente sur le poste.

Avertissement : Utiliser avec précaution, ca va fermer les logiciels en court d'utilisation.

Vous pouvez utiliser a la place *Lancer les installations en attentes pour les applications non lancées* Pour éviter toute perte de travail

30.4 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.

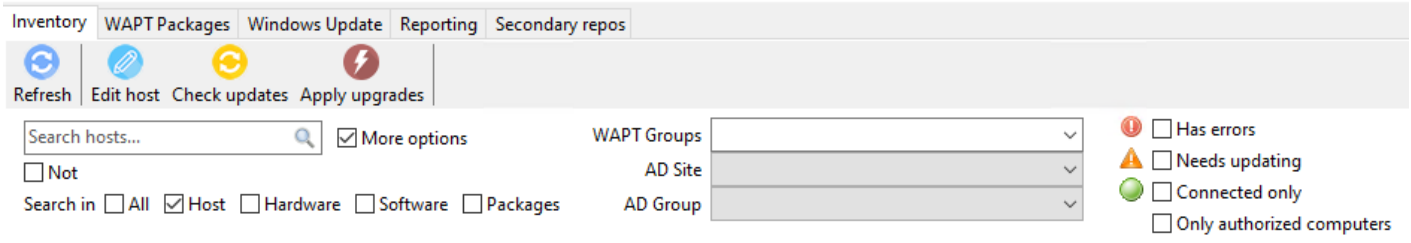


FIG. 2 – Les fonctionnalités de recherches avancées dans la console WAPT

Tableau 1 – Choix des filtres

Options possibles	Description
Host	La section *Host* dans l'onglet *Hardware inventory* quand un hôte est sélectionné
Hardware	La section *DMI* dans l'onglet *Hardware inventory* lorsqu" un hôte est sélectionné
Software	La section *Software inventory* lorsqu'un hôte est sélectionné
Package	Liste des paquets installés sur les hôtes sélectionnés
Has errors	Rechercher uniquement les hôtes dont les tâches ne se sont pas correctement terminées
Needs updating	Rechercher uniquement les hôte nécessitant une mise à jour
Connectés seulement	Rechercher uniquement les hôtes connectés
Seuls les ordinateurs autorisés	Rechercher uniquement les hôtes authorisés par le certificat de l'utilisateur en cours
WAPT Group	Filter les hôtes basés sur leur appartenance / dépendance à un paquet groupe WAPT
AD Site	Filter les hôtes basés sur leur appartenance / dépendance à Site et Services Active Directory
AD Group	Filter les hôtes basés sur leur appartenance / dépendance à un groupe Active Directory

Indication : Les filtres fonctionnent avec des expressions régulières.

30.5 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.

Fig. 3 – Affichage de l'inventaire depuis La console

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*).

More détail *here*.

30.6 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	**non**

suite sur la page suivante

Tableau 2 – suite de la page précédente

Nom	Multi-sélection
Vérifier les mises a jour	**yes**
Appliquer les mises à jour	**yes**
Appliquer les mises a jours d'applications qui ne sont pas lancées	**yes**
Proposes a l'utilisateur de lancer les mises a jours	**yes**
Envoie un message aux utilisateurs	**yes**
Lance les audits des paquets	**yes**
Ajouter un paquet aux dépendances de la machine	**yes**
Retire un paquets des dépendances de l'hôte	**yes**
Re-signe les paquets hôtes	**yes**
Ajouter un paquet dans les conflits de la machine	**yes**
Retire un paquet des conflits de l'hôte	**yes**
Supprime le poste	**yes**
Connexion en RDP	**non**
Assistance a distance	**non**
Mesh remote desktop	**yes**
Gestion de l'ordinateur windows	
Mettre a jour les GPO sur l'hôte	**yes**
Lance CleanMgr on host	**non**
Gestion de l'ordinateur	**non**
Gérer les utilisateurs et groupes	**non**
Gestion des services	**non**
Allumer avec WakeOnLan	**yes**
Redémarre les ordinateurs	**non**
Eteints les ordinateurs	**non**
Déclenche le scan des mises a jours Windows manquantes	**yes**
Déclenche le téléchargement des mises a jours Windows manquantes	**yes**
Déclenche l'installation des mises a jours Windows manquantes	**yes**
Rafraichis l'inventaire	**yes**
Lance un redémarrage de waptservice	**yes**

Note : For description of these actions, visit to *this documentation*.

30.7 Importe des paquets depuis un dépôt externe

30.7.1 Principe d'importation des paquets

Importer un paquet WAPT consiste à :

— Importing an existing WAPT package from an external repository.
— Changing its prefix (for example from *tis* to *my-prefix*).
— Re-signing the WAPT package with the *Administrator*'s or the *Code signing* private key to allow the deployment of the imported package on your WAPT equipped hosts.
— Finally, uploading it on the main WAPT repository.

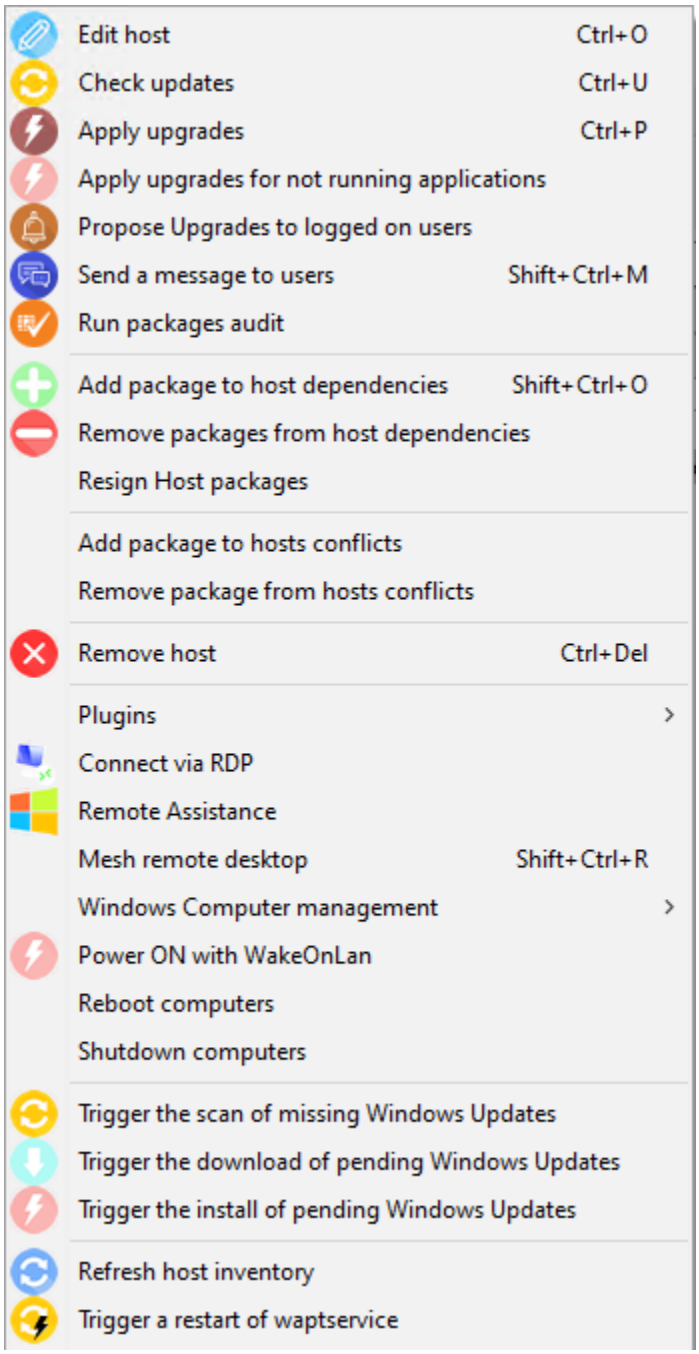


FIG. 4 – Menu de configuration de l'hôte

Attention : By importing a package in your repository and signing it, **YOU THEN BECOME RESPONSIBLE** for that package and for what it does. **It has been signed with your own private key**.

Tranquil IT disclaims any liability if you choose to use WAPT packages retrieved from its repositories. Without a support contract, Tranquil IT does not guarantee the suitability of the package for your own particular use case, nor does she guarantee the ability of the package to comply with your *Organization*'s internal security policies.

— Go to the *Private repository* tab.

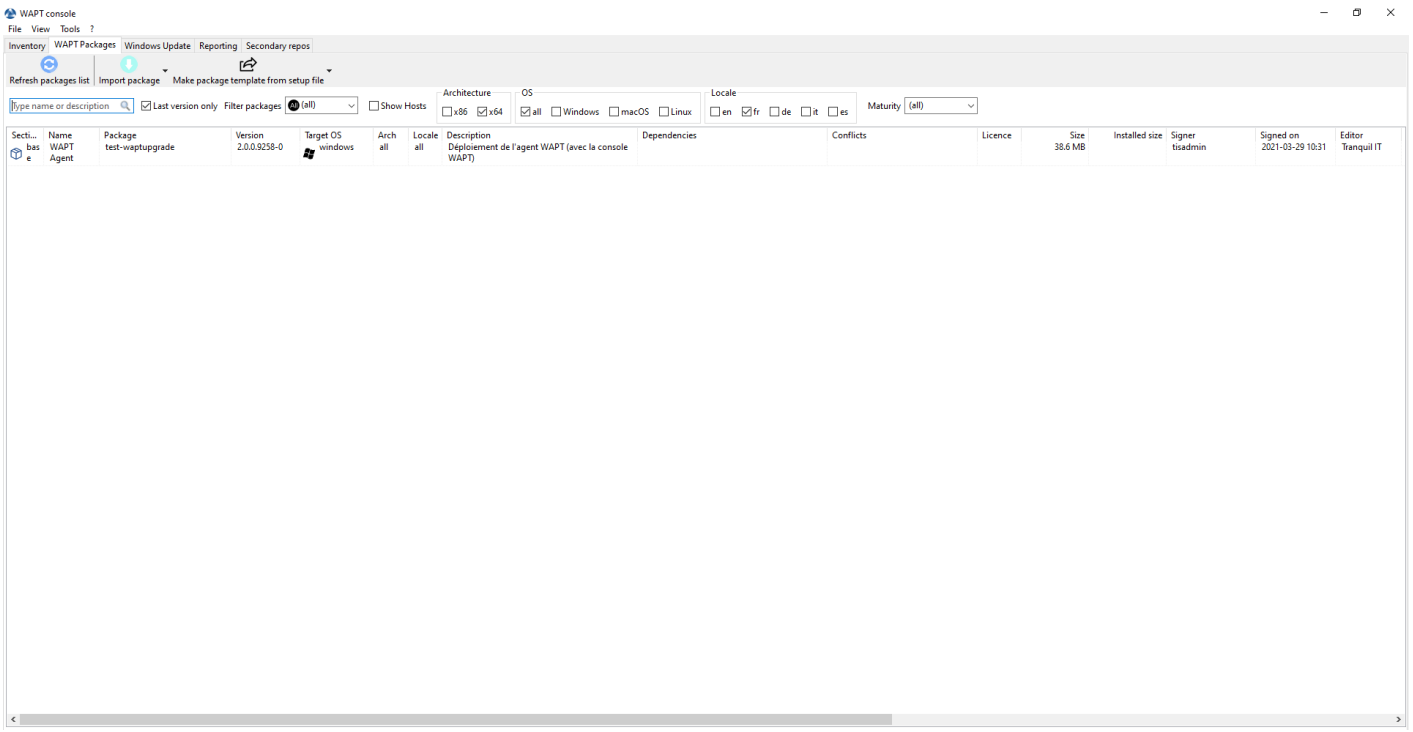


FIG. 5 – Les logiciels disponibles sont affichés dans la console WAPT

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*.

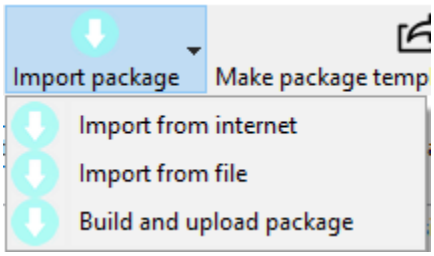
30.7.2 Importing a package from an external repository on the Internet

Cette première méthode vous permet de télécharger des paquets directement depuis un dépôt externe WAPT dans votre *Organisation*.

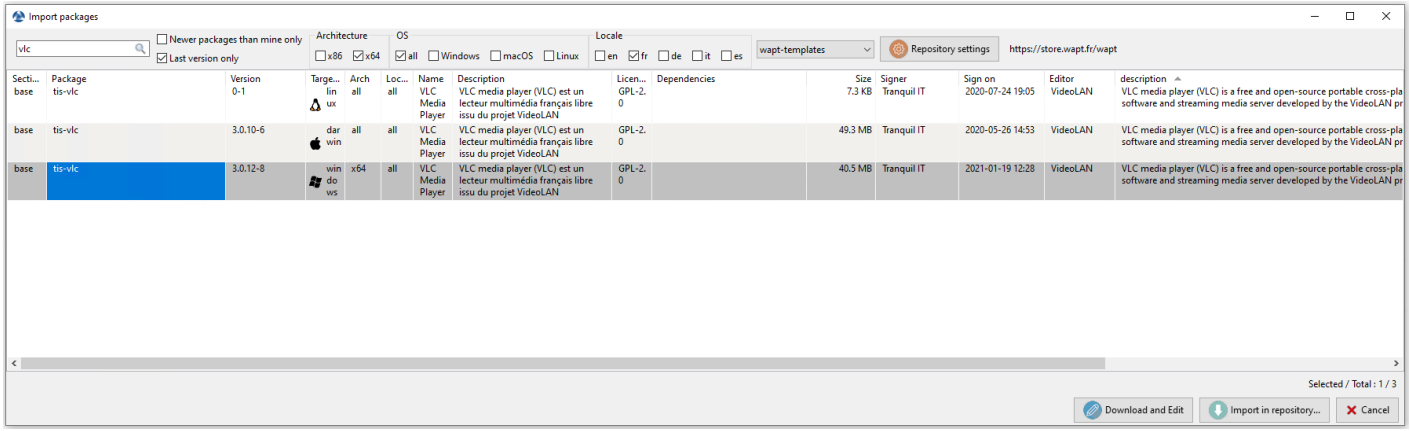
By default the Tranquil IT repository is configured, to add another repository check the documentation for *configuring the external repositories* ;

Note : By default, the SSL/ TLS certificates to external repositories are verified.

— Click on *Import package* and *Import from 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.



— There are 2 methods for importing a WAPT package :
  — pour importer un paquet, sélectionner un paquet puis *Clic-droit → Importer* ;
  — ou en bas a droit de la console *Importer dans le dépôt* :
— Validate the importation in your local repository.

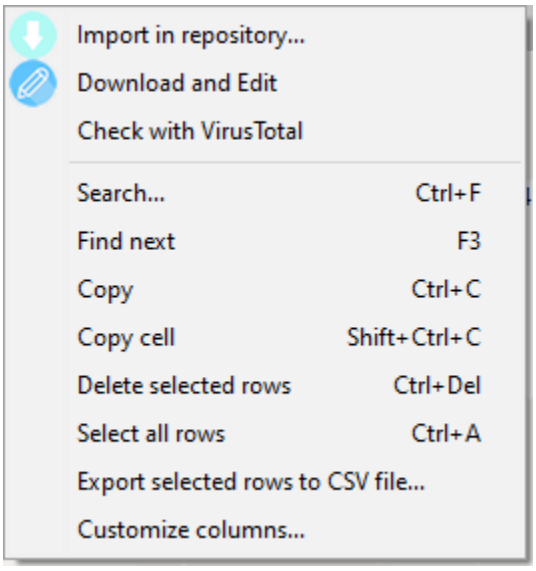
Note : It is possible to *change the maturity* of a WAPT package before importing the package into your private repository.

— The download of the package starts.

— Then, enter your private key password.

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.



wapt-resources/wapt_console-import-package-options-2.png

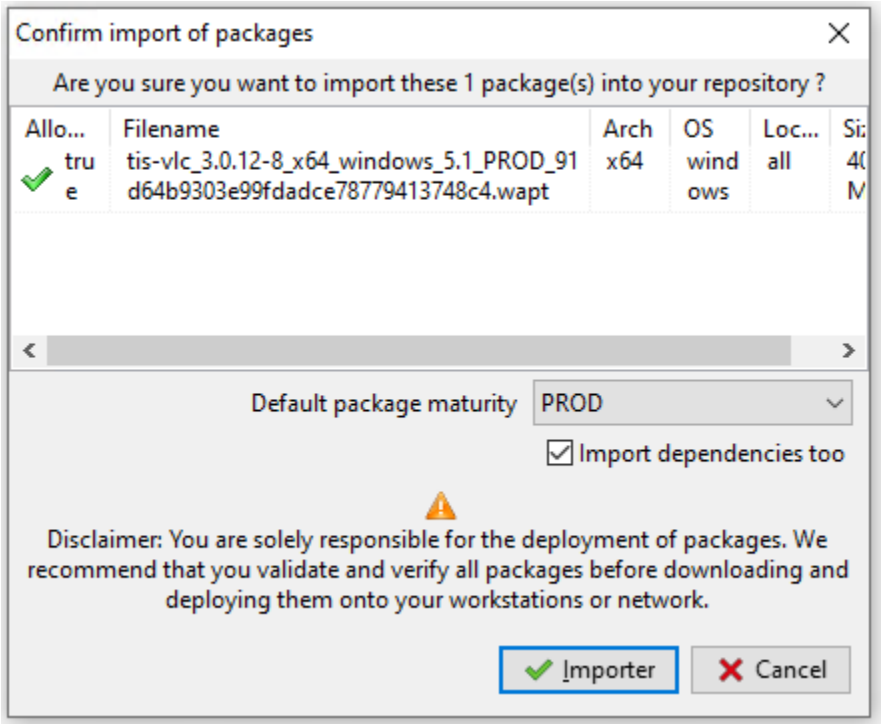


FIG. 6 – Confirmez l'import du paquet

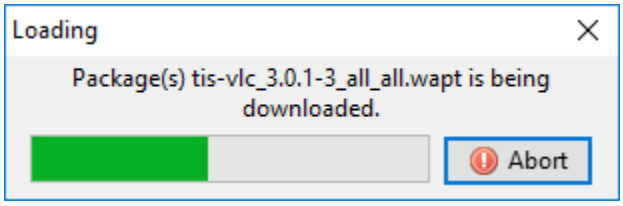


FIG. 7 – Processus de la progression de l'import du paquet

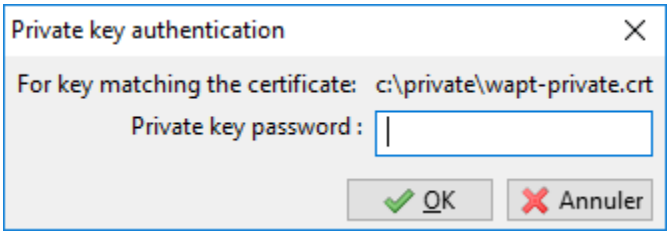


FIG. 8 – Entering the password for unlocking the private key

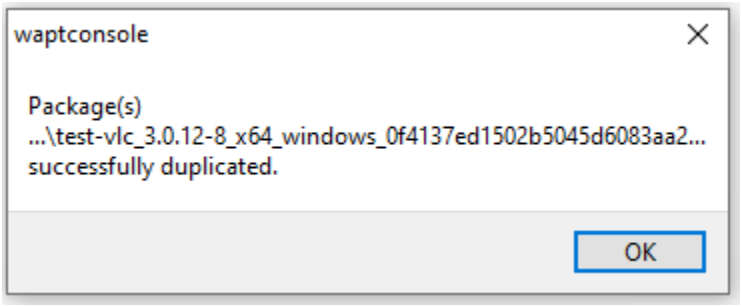


FIG. 9 – Confirmation de l'import réussie

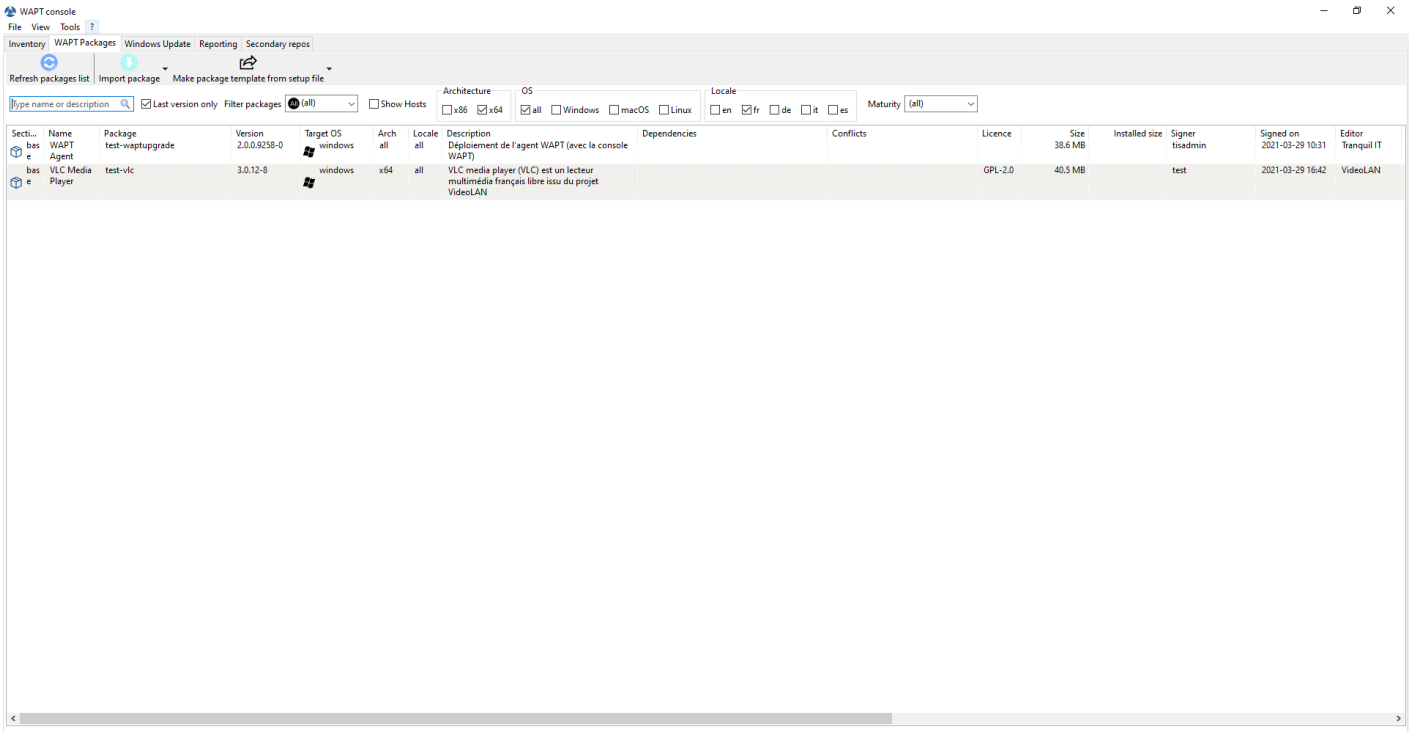


FIG. 10 – La console WAPT affiche le paquet importé

Changer la maturité d'un paquet WAPT avant de l'importer dans le dépôt

It is possible to change the maturity of a WAPT package before loading it into your private repository by choosing **DEV**, **PREPROD** or **PROD** in *Default package maturity*.

Editez un paquet avant de l'importer

It is possible to edit a package downloaded from an external repository before importing it into your main WAPT repository.

- Pour cela 2 choix disponible :
  - pour importer un paquet, sélectionner un paquet puis *Clic-droit → Importer* ;
  - or by clicking *Download and Edit* on the bottom right of the window ;

PyScripter, if installed, will open the `control` and `setup.py` files of the WAPT package.

For more information, visit the documentation on *creating WAPT packages from scratch*.

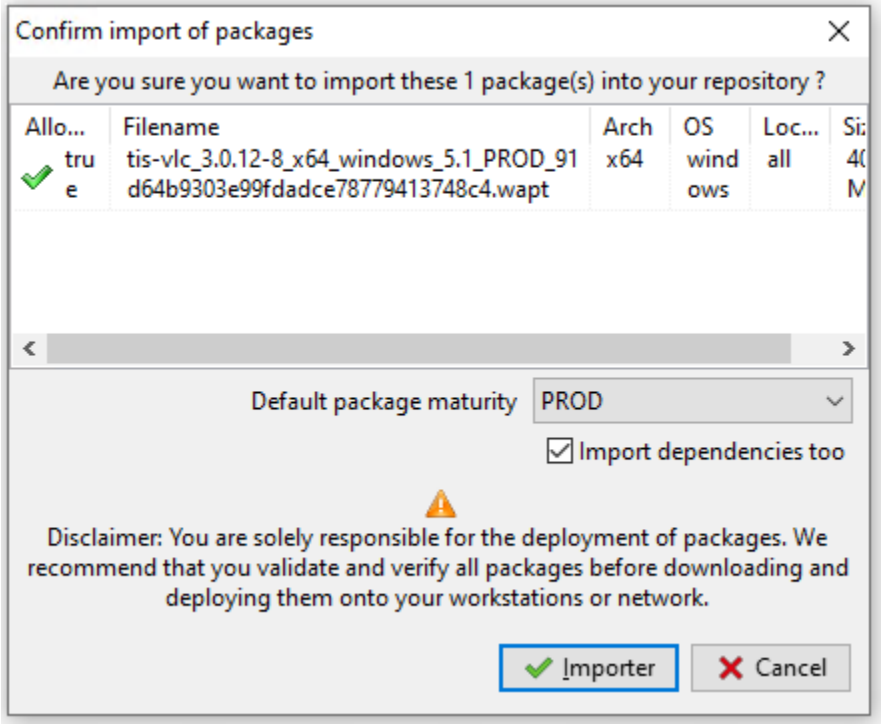


FIG. 11 – Choisir la maturité du paquet WAPT avant l'import

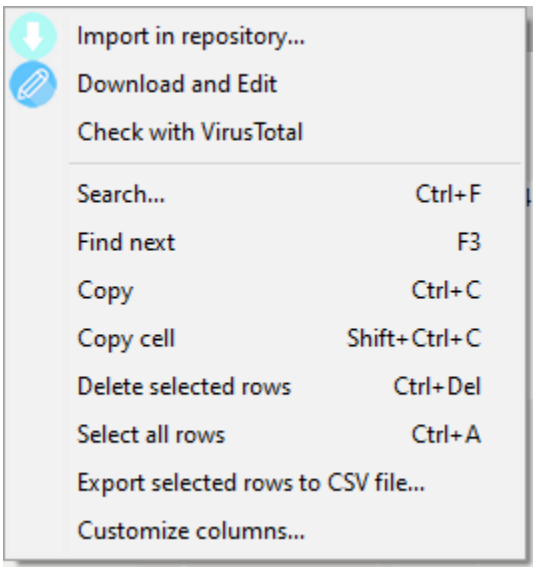


FIG. 12 – Download and Edit

wapt-resources/wapt_console-import-package-options-2.png

FIG. 13 – Download and Edit

30.8 Dupliquer des paquets depuis un dépôt externe

30.8.1 Importer un paquet WAPT depuis un fichier

You can import a `.wapt` file from any storage.

— Click on *Import package* and then *Import from file*.

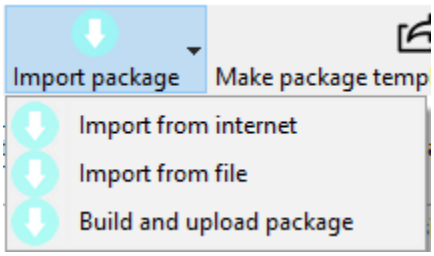


FIG. 14 – Importer un paquet WAPT depuis un fichier

— Select the file to import.

— Click on *Open* to import the file.

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.

30.8.2 Changer le préfixe et resigner un paquet WAPT

When uploading your new WAPT package to your private repository, the changing of the prefix and the re-signing of the WAPT package are transparent and automatic.

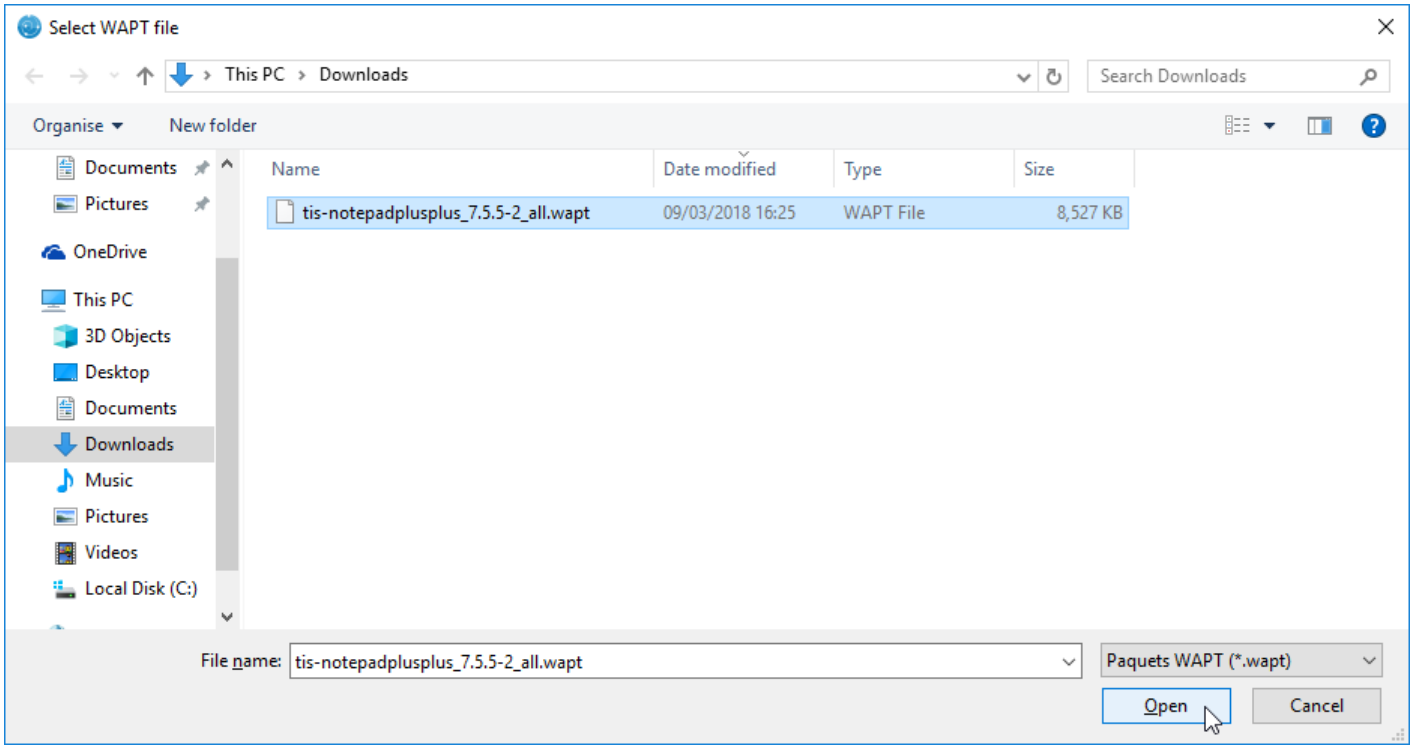


FIG. 15 – Sélectionnez le fichier à importer

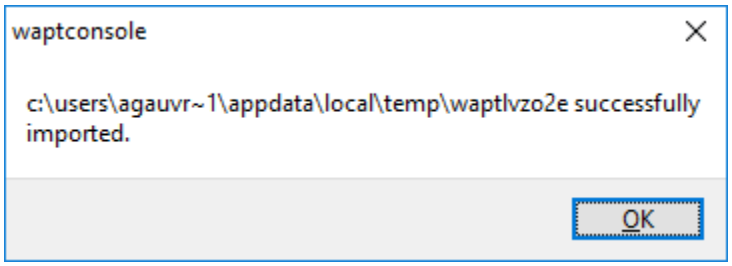


FIG. 16 – Fichier importé avec succès

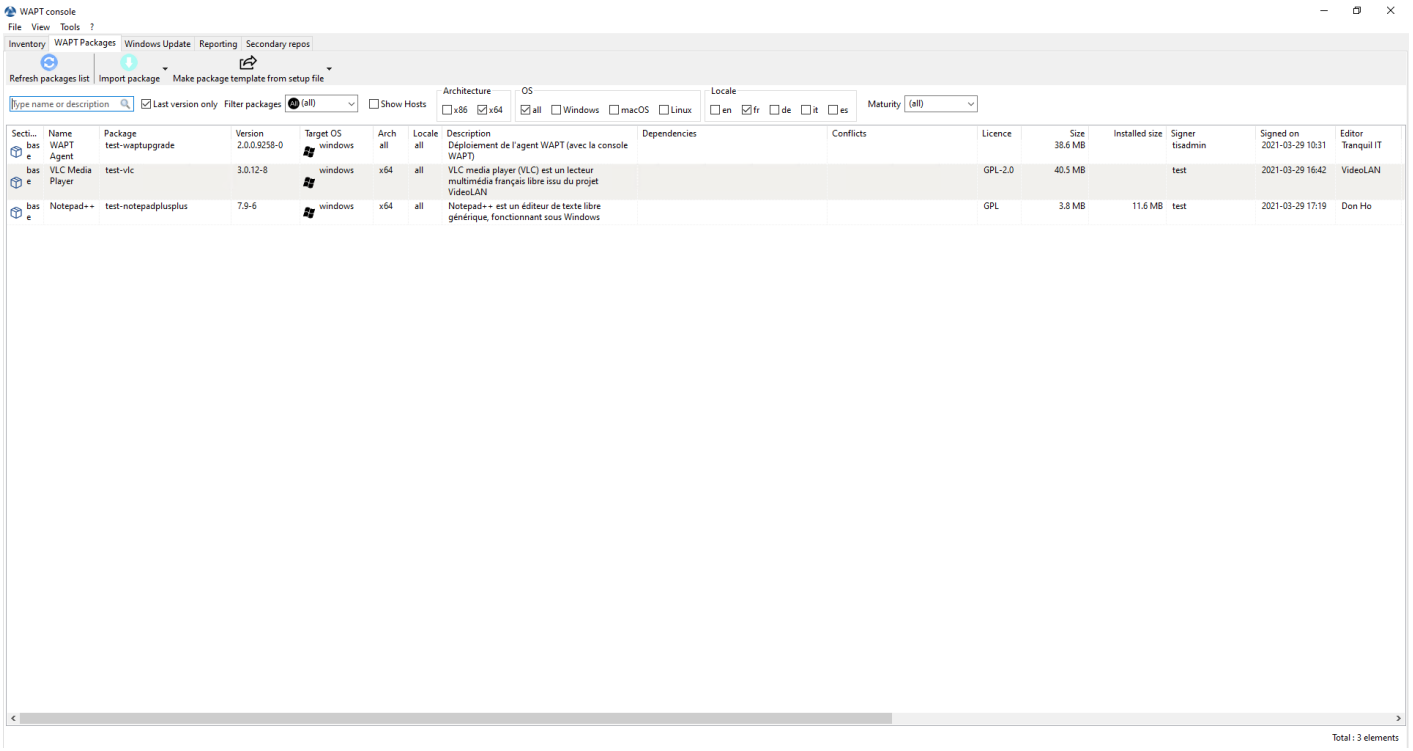


FIG. 17 – Le paquet WAPT importé s'affichera dans votre dépôt WAPT local

30.9 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*.

FIG. 18 – Changing the maturity of a WAPT package

30.9.1 Afficher tous les paquets

Pour afficher toutes les versions des paquets, décochez *Dernière version seulement*.

30.9.2 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* ;

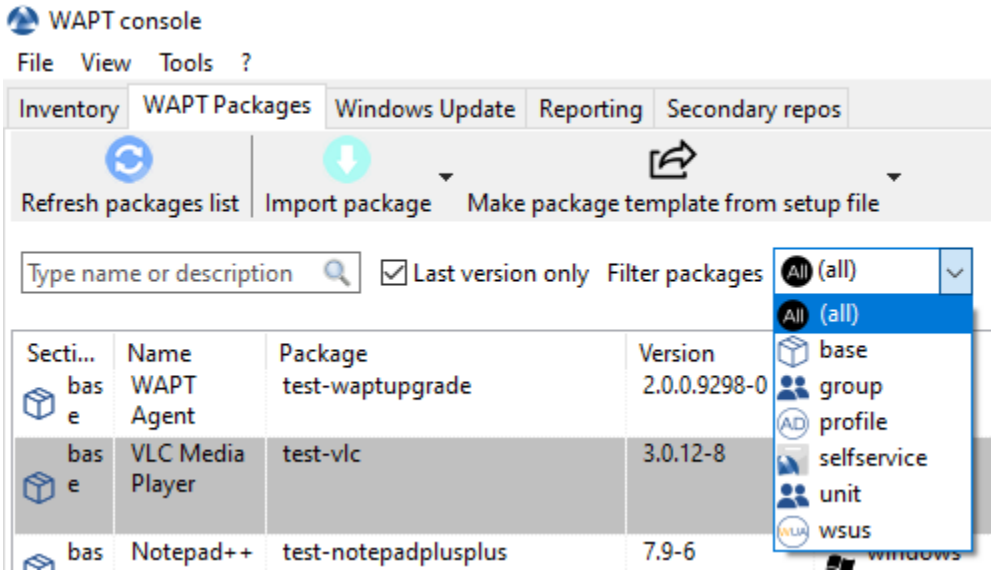


FIG. 19 – Filtrer par type de paquet WAPT

30.9.3 Autres filtres

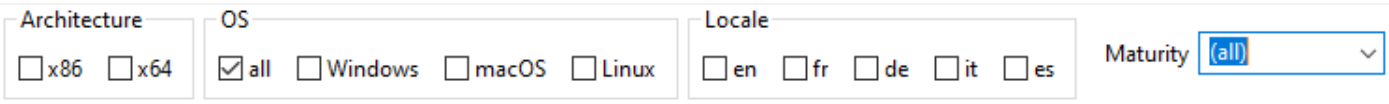


FIG. 20 – Filtrer avec d'autres attributs

Les autres filtres disponibles sont :
- architecture :
  - *x86* ;
  - *x64* ;

— OS (Operating System) :
  — *all* ;
  — *Windows* ;
  — *macOS* ;
  — *Linux* ;
— locale :
  — en ;
  — fr ;
  — de ;
  — it ;
  — es ;
— maturity :
  — PROD ;
  — PREPROD ;
  — DEV ;

30.9.4 Faire une recherche basé sur un paquet WAPT

In the repository, select the package and then click on *Show Hosts*.

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*).

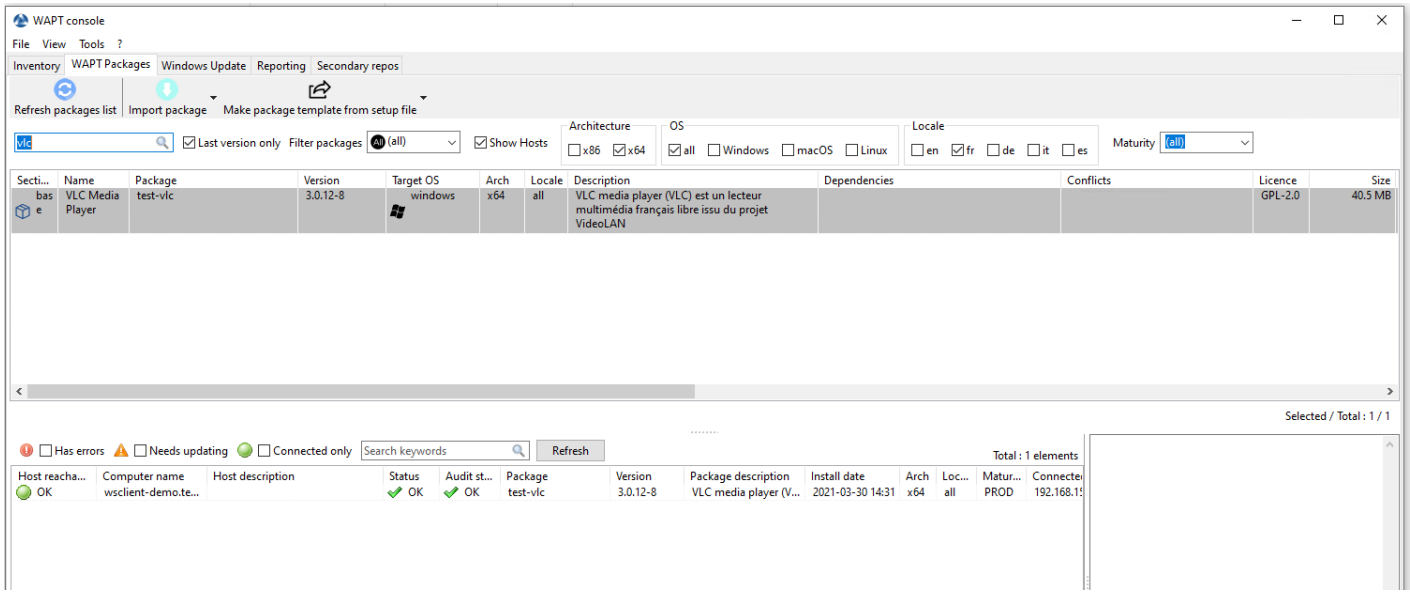
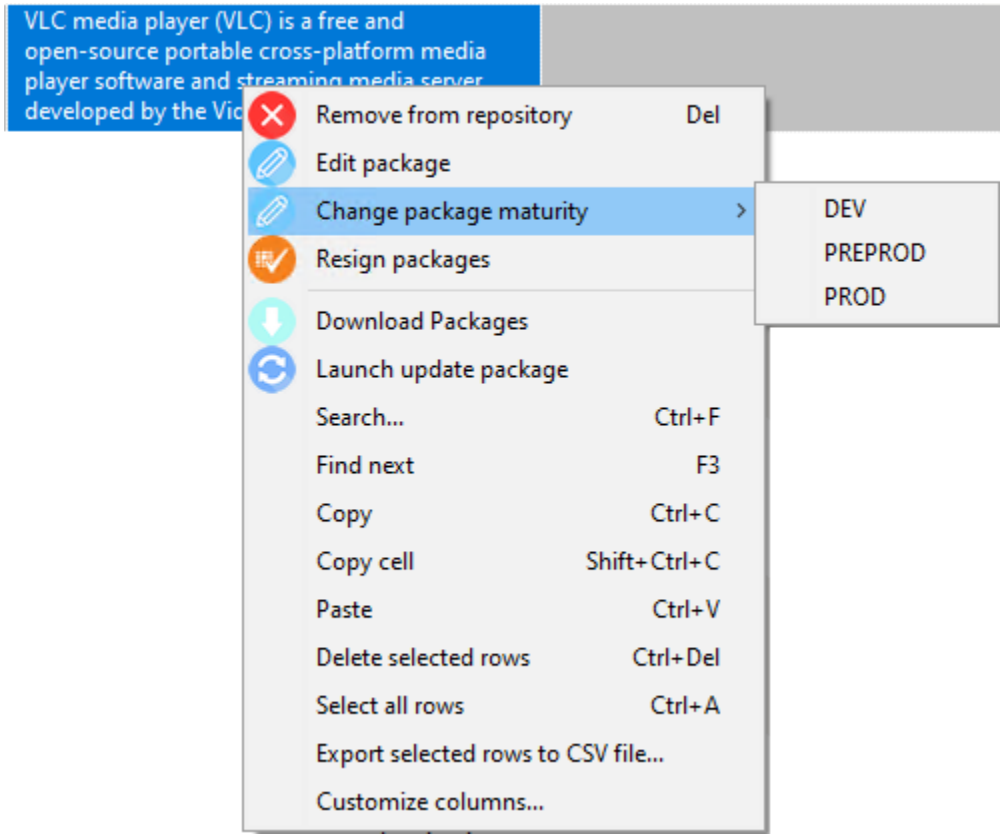


FIG. 21 – Filtrer par paquet

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.

30.9.5 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 choisis une fenêtre apparait :

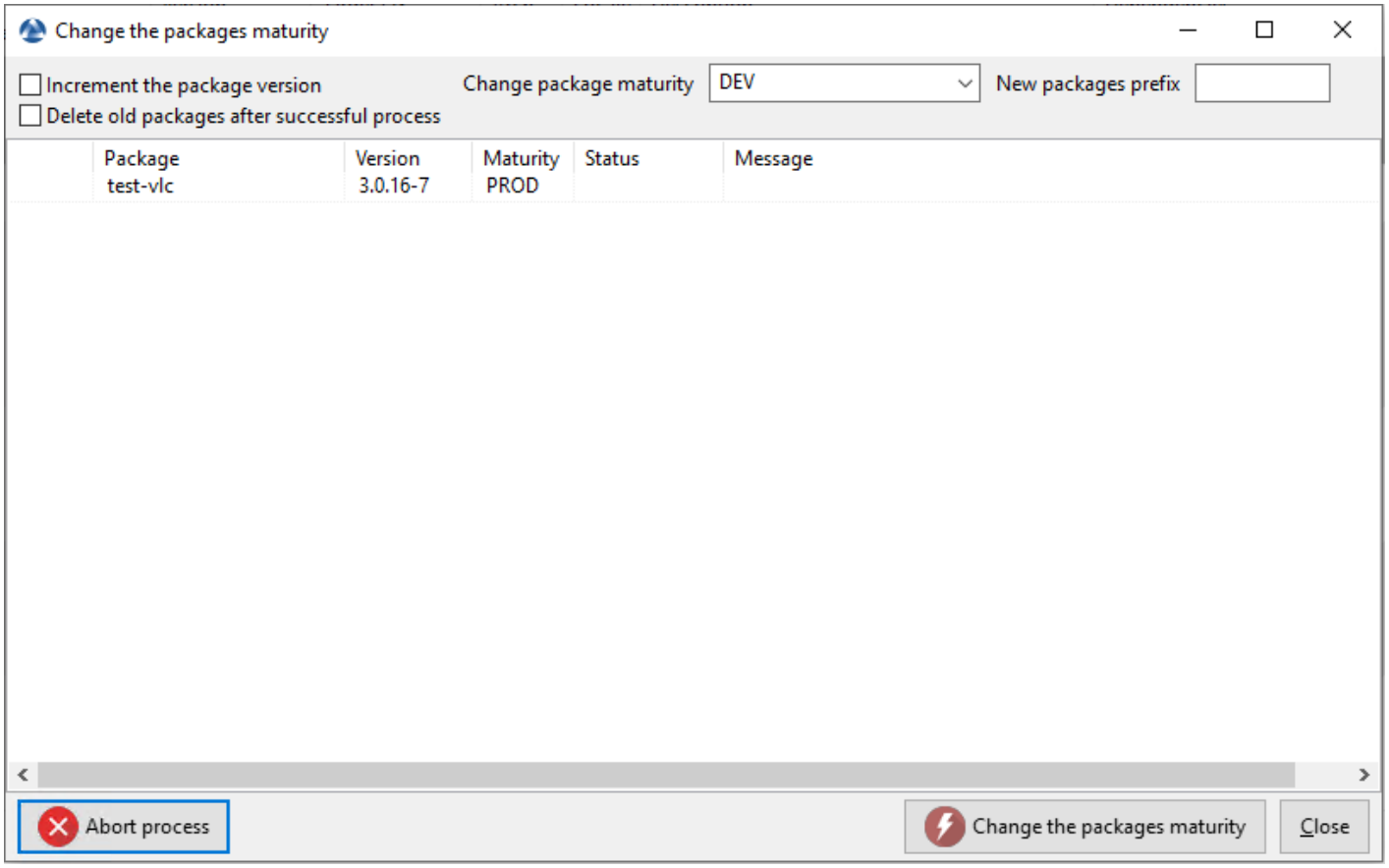
Plusieurs choix sont possibles :

Label	Description
Package'Increment the package version"	Increments the packaging version (version number after -).
Delete old packages after successful process	Delete the old WAPT package after having changed the maturity.
Change la maturité du paquet	Configure the new maturity of the WAPT package.
Prefix du nouveau paquet	Configure a new prefix for the WAPT package.

Note : You can stop the process by pressing the *Abort process* button.

Once finished, the status switches to ✅.

Indication : Changer la maturité de multiples paquets en une fois


Change the packages maturity
Increment the package version
Delete old packages after successful process
Change package maturity
DEV
New packages prefix
Package
Version
Maturity
Status
Message
test-vlc
3.0.16-7
PROD
Abort process
Change the packages maturity
Close

Avertissement : Changing the maturity of the package will change the hash of the package.

If the package is used in a GPO, like **waptupgrade**, you'll have to change the hash in your GPO.

30.9.6 Créer un groupe de paquet

Group packages allow you to create a package containing other packages to be affected as dependencies.

To create a group of packages, go to the *WAPT Packages* tab, then click on the *Make package template from setup file* button and finally choose *Group*.

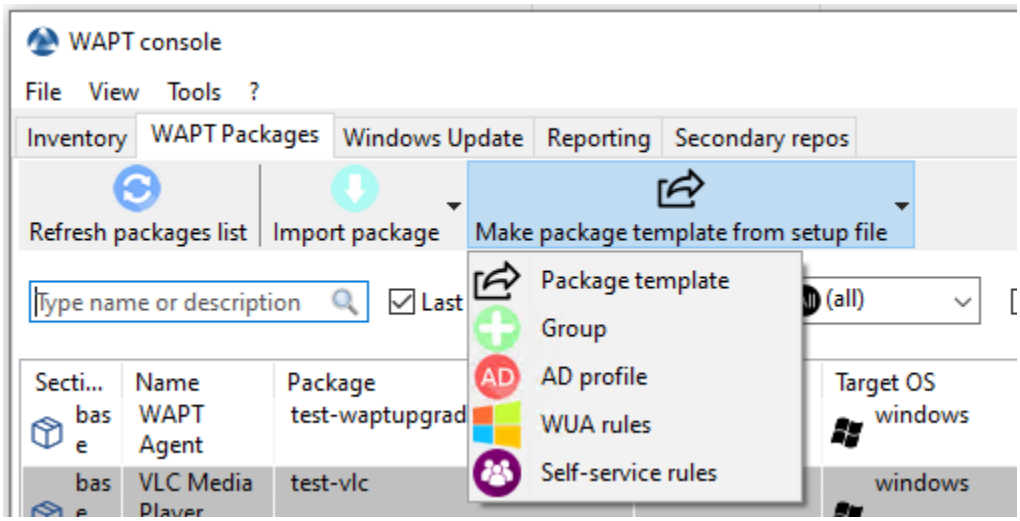


FIG. 22 – Grille de groupe de paquet

— Change the name in *Package name*.
— Fill in the description.
— Add packages to the group package by dragging and dropping them or by Right-clicking on the package name, and adding it to the bundle.
— Click on *Save* to save the bundle.

Indication : To uninstall a package, it is possible to add is as a forbidden package to a package group. The forbidden package, if installed, will be removed before other packages are installed.

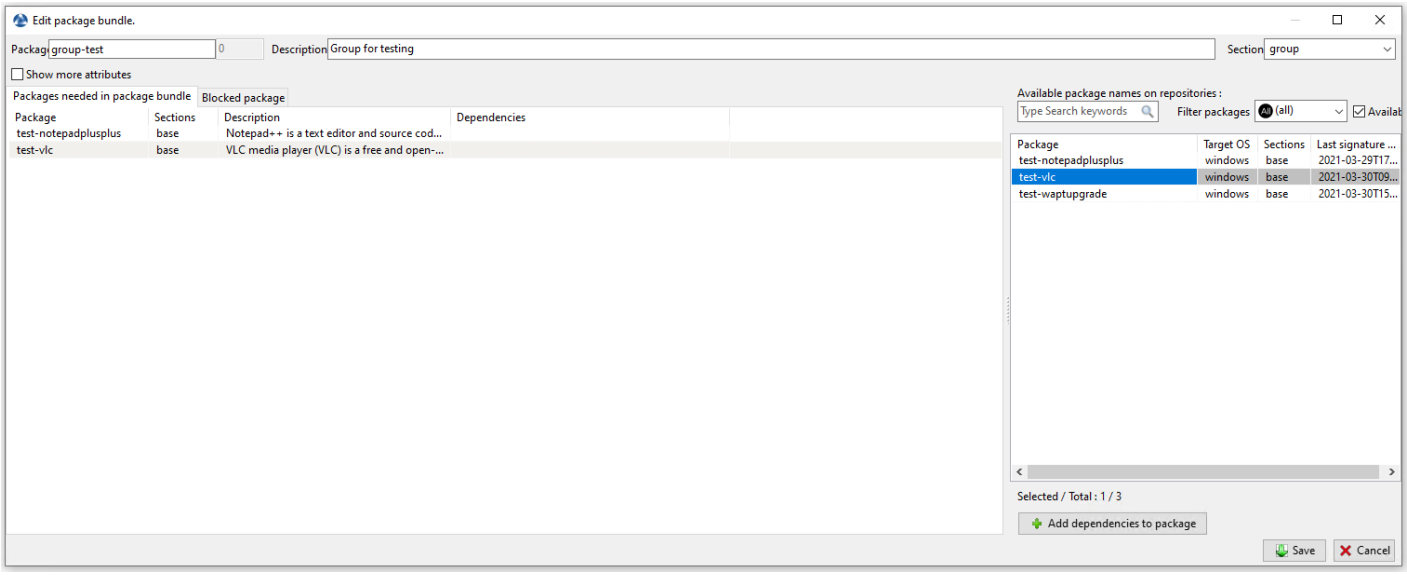


FIG. 23 – Créer un groupe de paquet

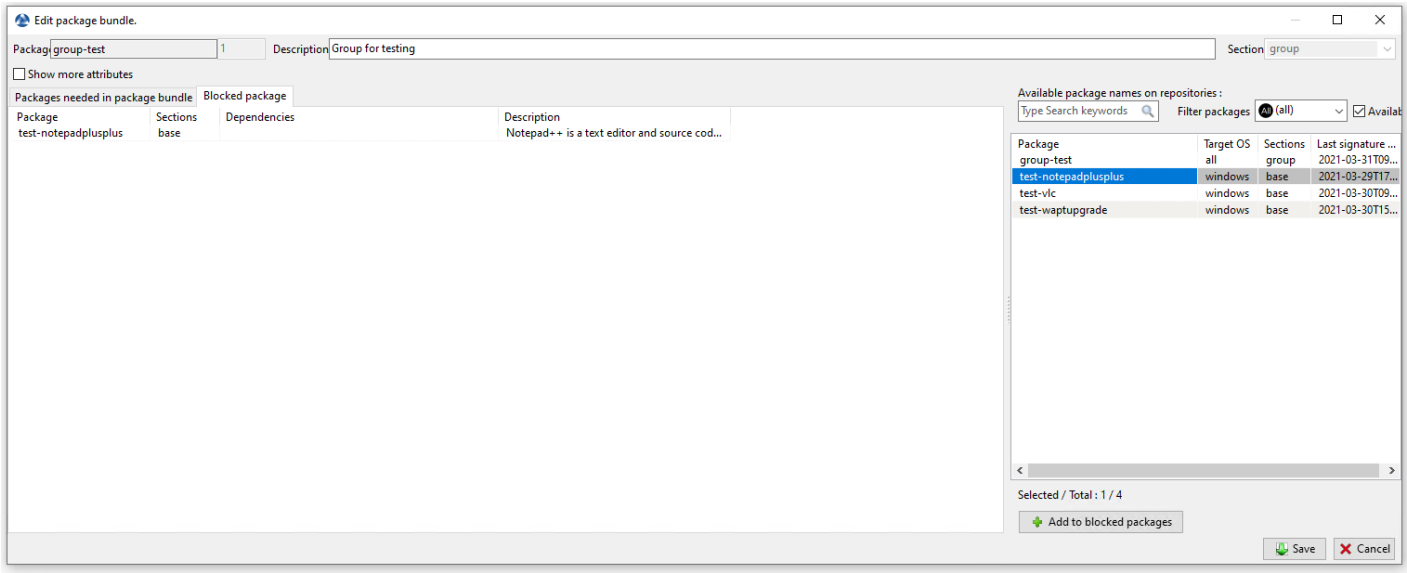
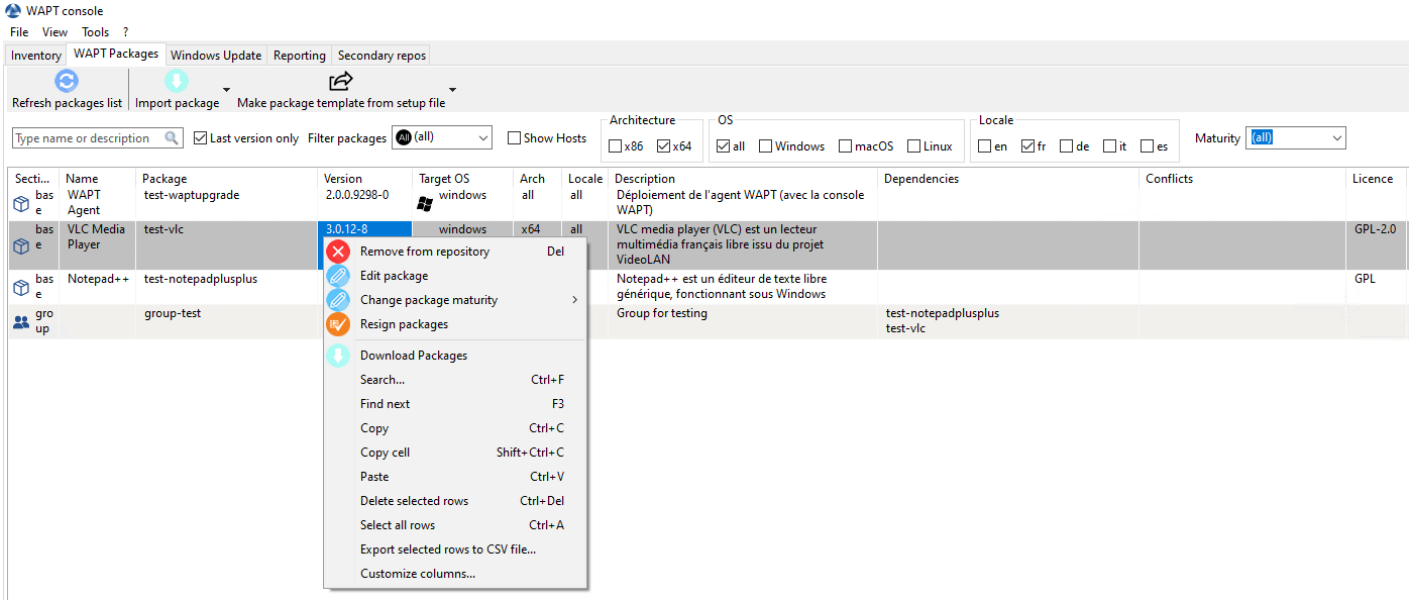


FIG. 24 – Adding a forbidden package to a host

30.9.7 Supprimer un paquet WAPT



To delete a package from the repository, do a *Right-click → Remove from repository*.

Indication : You can select multiple packages at once to delete.

30.9.8 Editer un paquet WAPT

To edit a package, do a *Right-click → Edit package*.

The package will be downloaded locally in **the base package development folder**, set in the *console settings*.

If **`PyScripter`** is installed, it will automatically open the `control` and `setup.py` files.

Once edited you can upload the package *using the WAPT console*.

30.9.9 Déployer des paquets WAPT puis la console WAPT

You can deploy packages on hosts using multiple methods :

- Directly by *adding a WAPT package to the selected host(s)*.
- By *adding a WAPT package to an Organizational Unit* WAPT of which the host is a member.
- By *adding a package to a host profile* WAPT that is applied to the host.
- By *adding the package to a group package* of which the host is a member.

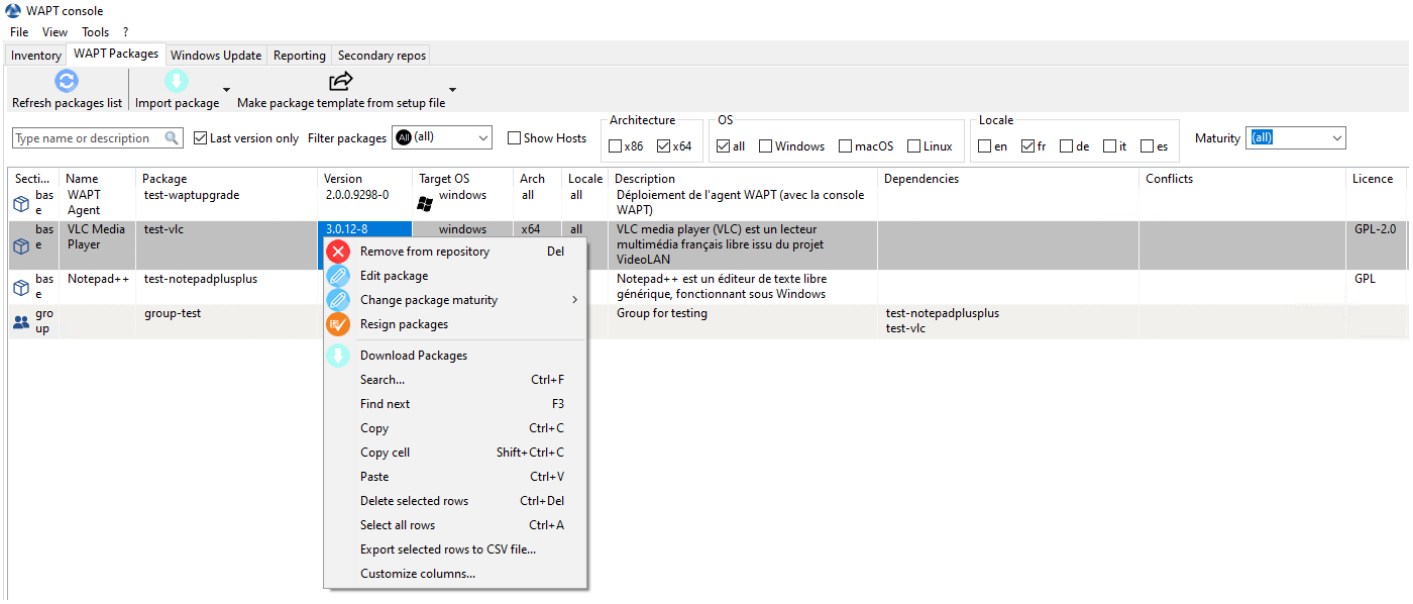


FIG. 25 – Editer un paquet WAPT

CHAPITRE 31

Utiliser la console WAPT avancé

Cette page détaille l'utilisation avancée de la console WAPT.

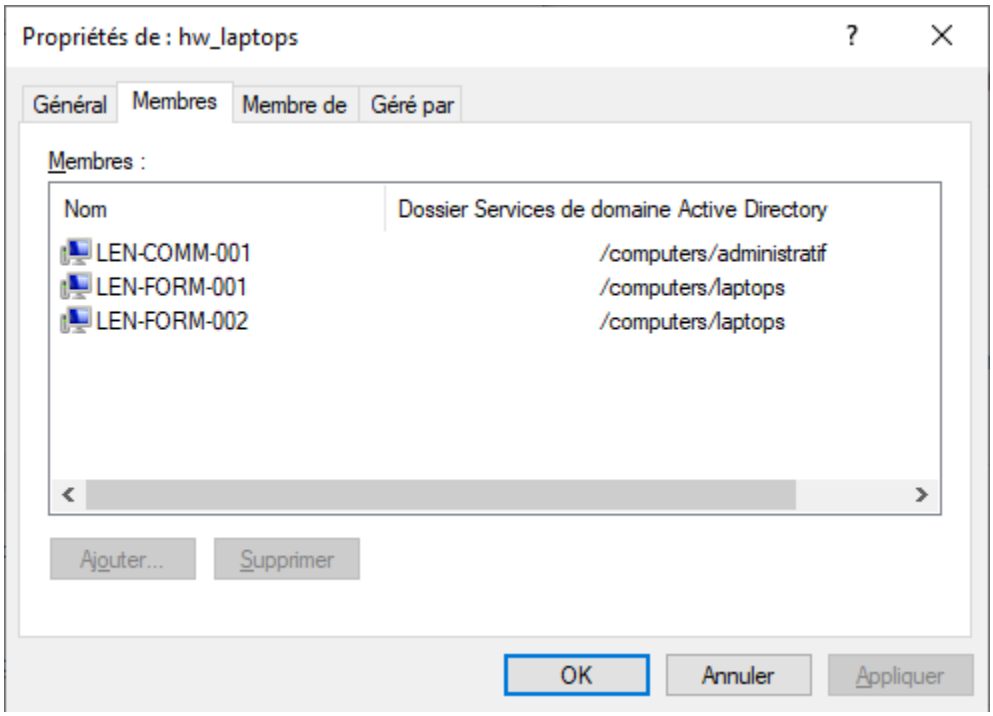
31.1 Utiliser des paquets profile dans WAPT

31.1.1 Principe de fonctionnement

WAPT Enterprise propose une fonctionnalité paquet profile Active Directory.

Cela automatise l'instalaltion du logiciel WAPT ainsi que ses paquets de configuration sur des hôtes, basé sur leur appartenance aux Groupes de SecuritéOrdinateur Active Directory.

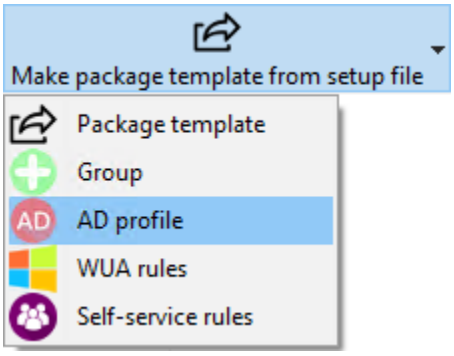
Important : Les groupes de Securité Ordinateur Active Directory contiennent des Ordinateurs et non pas des Utilisateurs.



Avertissement : Automatically installing software and configurations based on user and user group membership is not implemented with WAPT and such implementation is not desirable. The use case of installing oftware based on user profile is better served with the differentiated *self-service* feature that is also available with WAPT Enterprise.

31.1.2 Créer des paquets profile dans la console WAPT

You can create *profile* bundle packages by clicking on *Make package template from setup file -> AD profile*.



Important : Pré-requis :
- The *profile* package name must be **exactly** the same as the AD Security group name.
- The *profile* package **name is case sensitive**.

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.

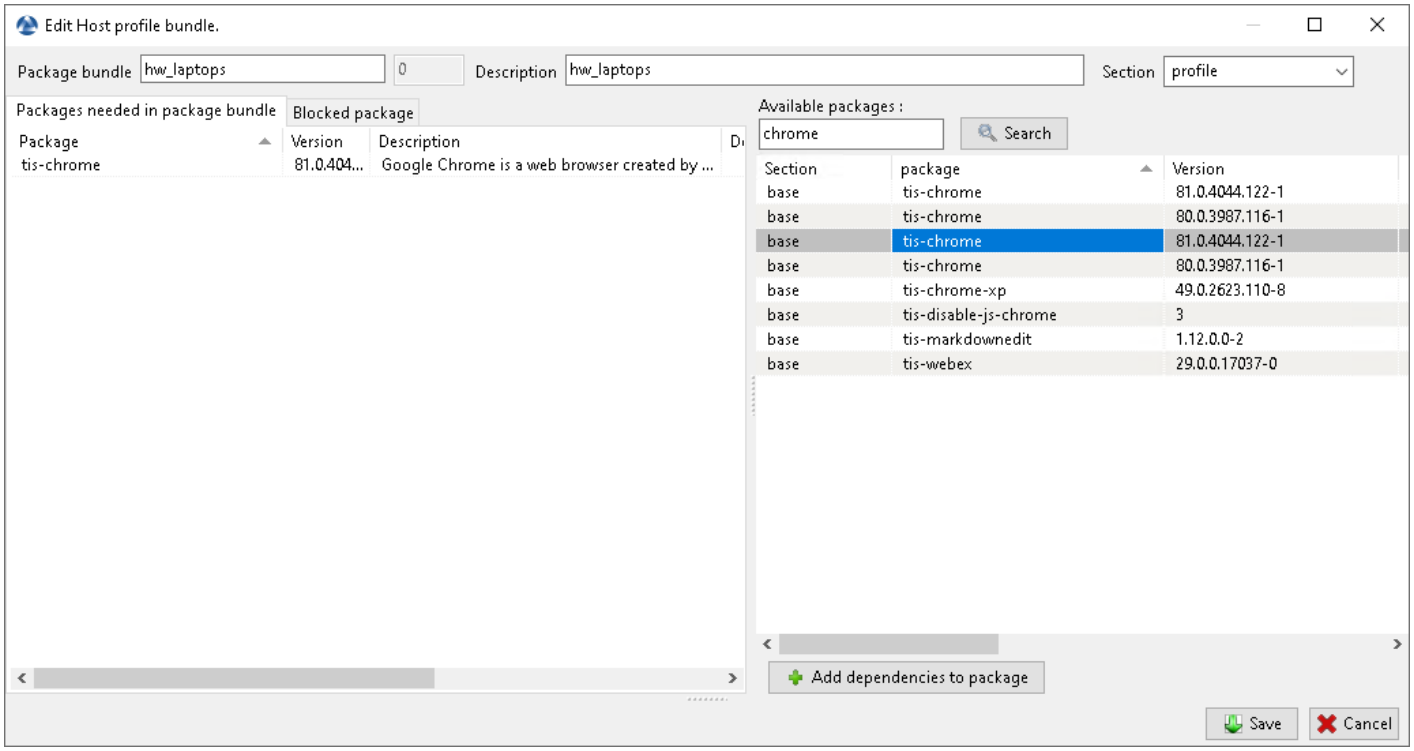


FIG. 1 – Ajouter un paquet au paquet profile

Sauvegardez le paquet *profile* qui sera alors téléversé vers le serveur WAPT.

31.2 Utiliser des Unités Organisationnelles dans WAPT

31.2.1 Principe de fonctionnement

WAPT Enterprise offers Organizational Unit packages functionality.

It automates software installations based on your Active Directory organization.

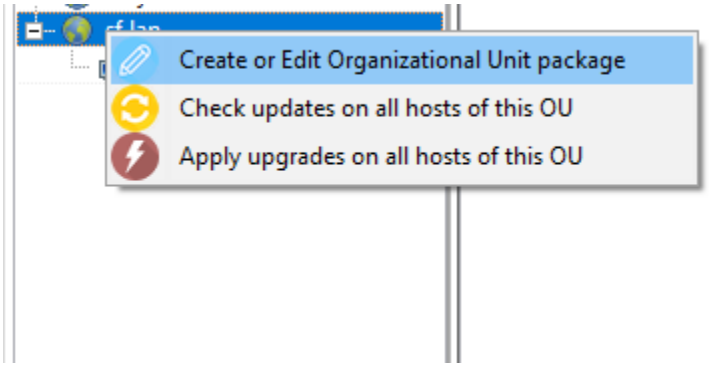
L'agent WAPT connait son emplacement dans l'arborescence Active Directory, il connaît donc pour cette raison la hiérarchie des Unités Organisationnelles qui le concerne, par exemple :

```
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
```

If an Organizational Unit package is defined on each level, the WAPT agent will automatically download packages and configurations that are attached to each level. Using inheritance, WAPT will apply packages and dependencies that have been attached to that Organizational Unit.

31.2.2 Creating Organizational Unit packages

You can create *unit* packages by *Right-clicking on an OU → Create or Edit Organizational Unit package*.



Une fenêtre s'ouvre et vil vous ai demandé de choisir les paquets qui doivent être inclus dans le paquet **unit**.

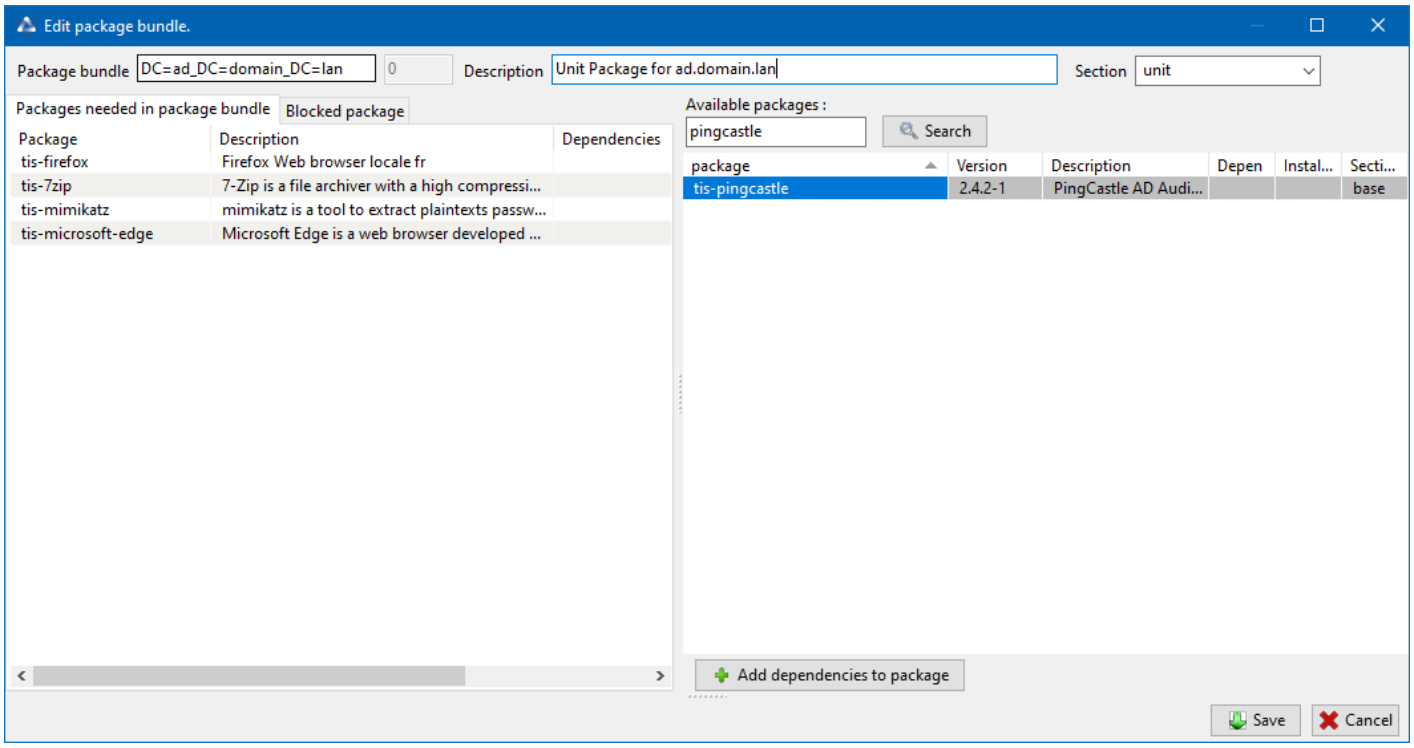


FIG. 2 – Ajouter un paquet au paquet unit.

Save the package and it will be deployed to all hosts belonging to that OU.

31.2.3 Actions available with Organizational Units

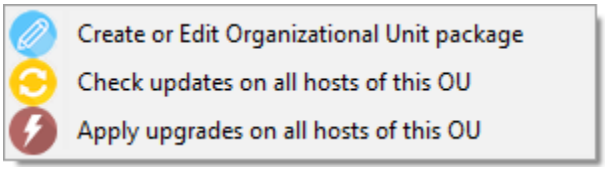


Fig. 3 – WAPT console showing options applicable to OU

You can see in the picture that *update* and *upgrade* actions can be performed through this menu, thus selecting hosts by their Organizational Unit.

Creating or Editing Organizational Unit package

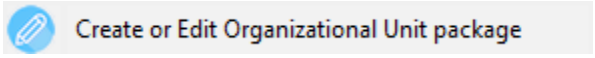


Fig. 4 – Creating or editing OU packages

Visit this documentation for more details on Creating or editing OU packages.

Checking updates on all hosts of this OU

Fig. 5 – Checking updates for hosts in the OU

This button will execute 2 actions on all hosts in the OU :

1. Give current state of the host to the server.
1. Server displays if the host must get updates.

Applying upgrades on all hosts of the OU

This button allows to apply waiting updates on the all hosts in the OU.

Indication :

You may filter how hosts are displayed based on the Active Directory
OU they belong to.

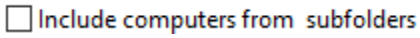


Fig. 7 – Including hosts in subfolders

The checkbox *Include hosts in subfolders* allows to display hosts in subfolders.

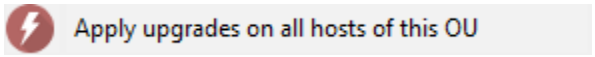


Fig. 6 – Applying upgrades to hosts in the OU

31.2.4 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 spécifité, 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 :

- You can manage these hosts with WAPT as if they where joined to the AD.
- **Out-of-domain and workgroup hosts are now showing up**
  in a familiar Active Directory tree view.
- *Unit* packages are usable on these hosts.

To setup a *fake* Organizational Unit on hosts, create an *empty WAPT package*, then use the following code :

```
# -*- 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
```

The `host_organizational_unit_dn` will be like below in `wapt-get.ini` :

```
[global]
host_organizational_unit_dn=OU=TOTO,OU=TEST,DC=DEMO,DC=LAN
```

Note :

- **Stick to a specific case with your `host_organizational_unit_dn`**
  (don't mix « dc »s and « DC »s, « ou »s and « OU »s…).
- Follow the case used in the DN/`computer_ad_dn` fields in the hosts grid.

31.3 Utiliser les WAPT Windows Update Agent (WAPTWUA)

WAPT is able to manage Windows Updates on your endpoints and replace automatic Windows update or a WSUS Server.

Note :

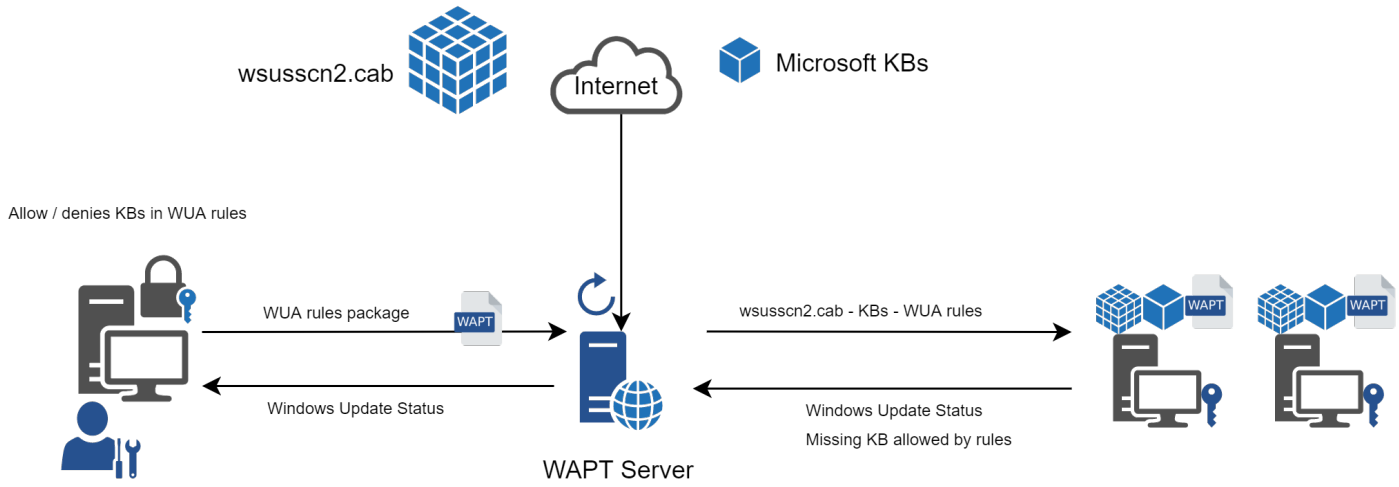
The internals of WAPTWUA is based on the WUA (Windows Update Agent) API.
For more information : https://docs.microsoft.com/en-us/windows/win32/wua_sdk/using-the-windows-update-agent-api

---

31.3.1 Principe de fonctionnement

Each Patch Tuesday (Patch Tuesday is an unofficial term used to refer the second Tuesday of each month when Microsoft releases software patches for their software products.), the WAPT server downloads an updated `wsusscn2.cab` file from Microsoft servers.

By default, downloads happen once a day and no download is triggered if the `wsusscn2.cab` file has not changed since the last download.



The `wsusscn2.cab` file is then downloaded by the WAPT agent from WAPT server repository and then passed on to WUA Windows utility to crunch the update tree for the host.

Regularly, the host will analyze the available updates using the `wsusscn2.cab` file. The host will send its list of needed updates to the WAPT server.

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 joursdemandées depuis les serveurs officiels Microsoft.

Indication :

This mode of operation allows WAPT to download only the necessary updates
on the computers, thus saving bandwidth, download time and disk space.

Note : On the WAPT server, downloaded updates are stored :
— On Linux hosts in `/var/www/waptwua`.

— On Windows hosts in `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` :

— **In case of repository replication, it is fully operational**
with WAPT Windows Update to reduce bandwidth use.

— **Do not forget to synchronize the `waptwua` folder**
if you are replicating your packages with distant repositories.

Note : Si dans votre compagnie un proxy est requis pour sortir sur Internet, alors assurez-vous d'avoir *confguré le serveur proxy dans le fichier waptserver.ini*.

31.3.2 Les différences entre WAPT Windows Updates et WSUS

WSUS downloads by default the updates for selected categories. This can lead to a very large update database and lots of storage.

WAPT Windows Update only downloads updates that have been requested by at least one client computer. This helps to keep the local database small (a few 10s of Gigabytes) and it can be easily cleaned up if you want to recover space.

31.3.3 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 1803 vers Windows 10 1903.

Major version upgrades are not handled in the same way as minor OS upgrades. Major upgrades are handled via the downloading of the new install ISO content (same content as for a fresh install) and running the **`setup.exe`** with the correct parameters. This process is the same for WSUS, SCCM and 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.

31.3.4 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.

31.3.5 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.

31.3.6 Configurer WAPTWUA sur l'agent WAPT

WAPTWUA is configured in `wapt-get.ini` in `[waptwua]` section.

Vous aurez alors plusieurs options :

Tableau 1 – Configuration options in the [waptwua] section in the wapt-get.ini

Options / Valeures par défaut	Description	Exemple
enabled = False	Enable or disable WAPTWUA on this host.	enabled = 1
direct_download = False	Télécharger les mises à jour directement depuis les serveurs Microsoft.	direct_download = 1
default_allow = False	Set if missing update is authorized or not by default.	default_allow = 1
download_scheduling = None	**Set the Windows Update scan recurrence (Will not do anything** if *waptwua* package rule or wsusscn2.cab file have not changed).	download_scheduling = 1
install_scheduling = None	**Set the Windows Update install recurrence** (Will do nothing if no update is pending).	install_scheduling = 2h
install_at_shutdown = False	Install update when the machine will shutdown.	install_at_shutdown = True
install_delay = None	Set a deferred installation delay before publication in the repository.	install_delay = 15d
allowed_severities = None	**Define a severity list that will be automatically accepted** during a WAPT windows update scan. ex : *Important*, *Critical*, *Moderate*.	allowed_severities = Important

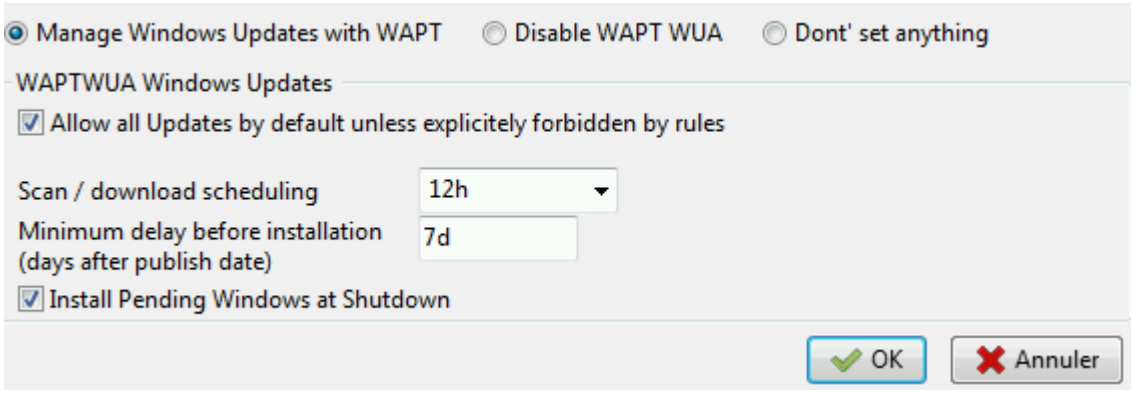
Indication : These options can be set when generating the WAPT agent.

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 :

Example source code to modify [waptwua] settings by package :




```
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')

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

31.3.7 Utiliser WAPTWUA depuis la console

The WAPTWUA are managed with two tabs in the WAPT console.

WUA Rules sub-tab in WAPT Package tab

The *WUA Rules* tab allows you to create *waptwua* rules packages.

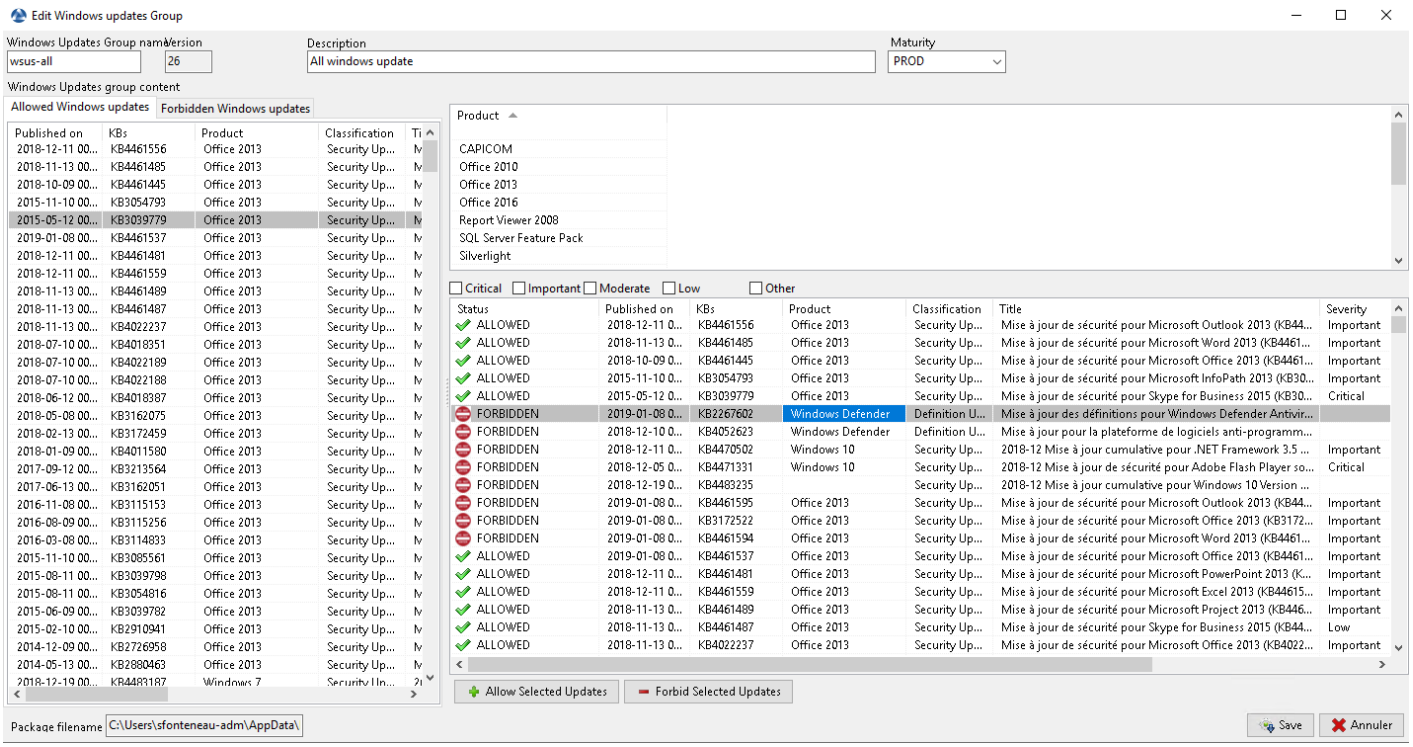
- **When this type of package is installed on a machine,**
  it indicates to the WAPTWUA agent the authorized or forbidden KBs (Knowledge Base articles).
- **When several *waptwua* packages are installed on a machine,**
  the different rules will be merged.
- **When a `cab` is neither mentioned as authorized,**
  nor mentioned as prohibited, WAPT agents will then take the value of `default_allow` in `wapt-get.ini`.

Note :

- If the WAPTWUA agent configuration is set to `default_allow = True`, then it will be necessary to specify the forbidden `cab`.
- If the WAPTWUA agent configuration is set to `default_allow = False`, then it will be necessary to specify the authorized `cab`.

Indication :

- To test updates on a small set of computers, you can set WAPTWUA default maturity to `PREPROD`.
- **You can then test the Windows Updates on a small sample of `PREPROD` hosts**
  and if everything is good, you can release the updates to the entire fleet of computers.



Windows Updates tab

The *Windows Update* tab lists all needed Windows Updates.

Important :

The server does not scan the `wsussc2.cab` itself,
: it lets the Windows Update Agent utility present on all Windows machines do it.

If an update seems to you as missing from the list, you must run a scan
: on one of the machines present in the console.

If you run a WUA scan on a Windows 10 agent, the CAB and Windows 10 files
: will be displayed on the *Windows Update* tab.

Le panneau de gauche affiche les catégories des mises à jour, vous permettant de filtrer par :

- Criticality ;
- Product ;
- Classification ;

Dans le panneau de droite, si la colonne *Téléchargée le* 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).

- You can force the download of an update by *right-clicking* → *Download*.
- You can also force the download of the `wsusscn2.cab` file with the *Download WSUSScan cab from Microsoft Web Site* button.
- You can see the Windows Updates download on the server with the *Show download task* button.

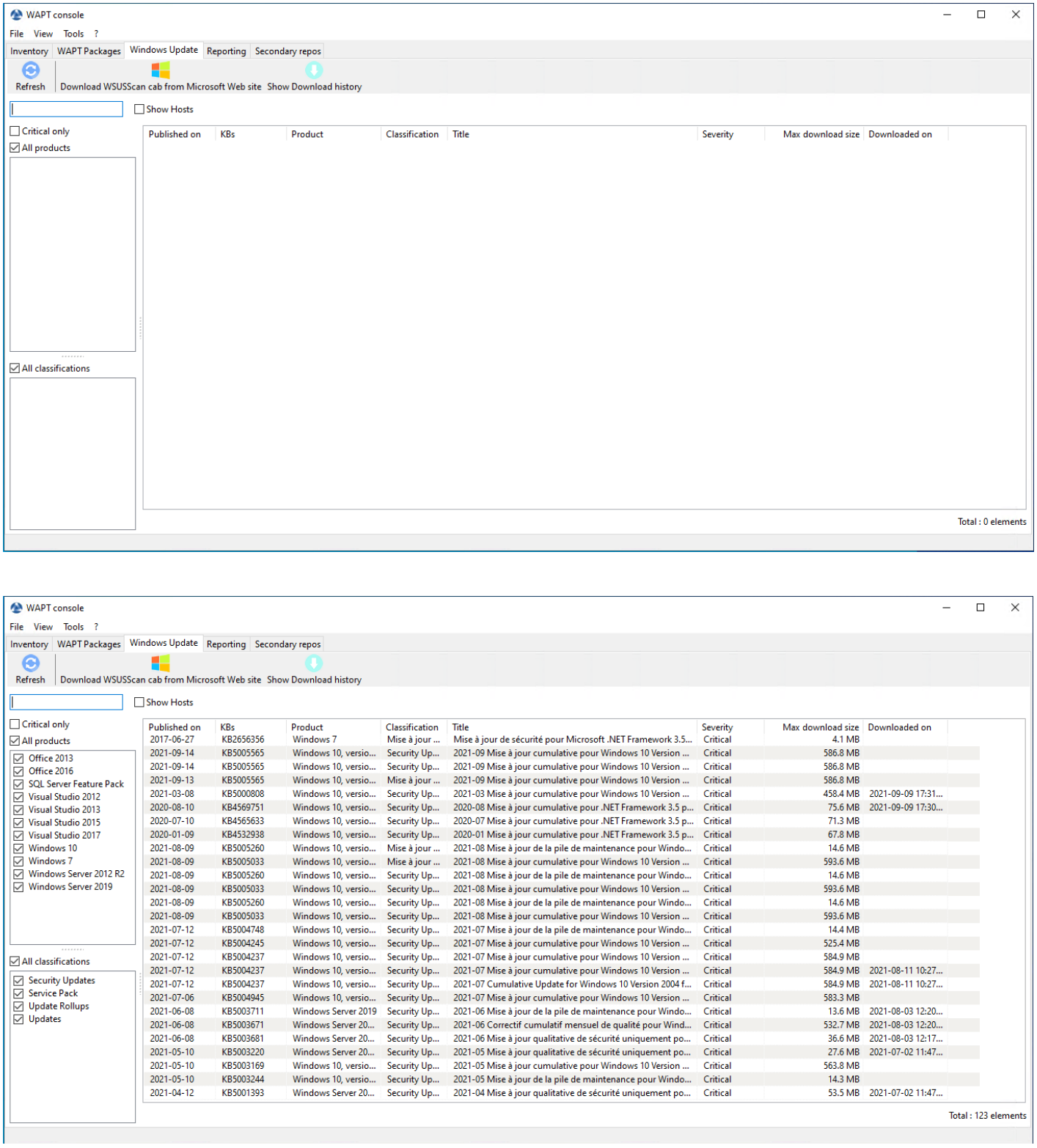


FIG. 8 – Liste des Windows Update

Indication : Toutes les 30 minutes, le serveur WAPT va chercher les mises à jour qui ont été demandées au moins une fois par les client WAPT et qui n'ont pas été téléchargées en mises en cache. » Si une mise à jour est en attente, le serveur WAPT var le télécharger depuis les sites officiels de Microfoft.

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*

Clean old Windows updates

To cleanup your `waptwua` folder, you can remove Windows Updates that are no longer needed.

WAPT server will only re-download deleted updates if one of the WAPT equipped hosts requests it.

On the WAPT Server, downloaded updates are stored :
- On Linux hosts in `/var/www/waptwua`.
- On Windows hosts in `C:\wapt\waptserver\repository\waptwua`.

Lancer WUA sur les clients

Depuis la console vous avez trois options.

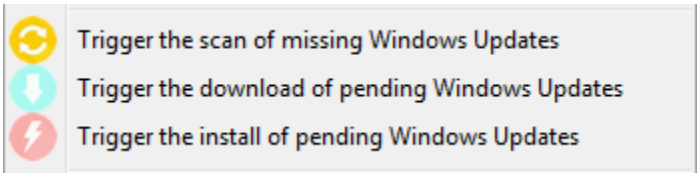


FIG. 9 – Les boutons d'action Windows Update disponibles depuis la console WAPT

- **The *Trigger the scan of pending Windows Updates* button**
  will launch the scan on the client and list all updates flagged for the OS.
- **The *Trigger the download of pending Windows Updates* button**
  will launch the downloading of pending updates on the client.
- **The *Trigger the install of pending Windows Updates* button**
  will launch the install of downloaded updates on the client.

Indication : When pending updates stored in cache need to be installed, the WAPT agent triggers the WUA service.

The WAPT agent will enable and start the WUA Service temporarily to install the updates. When updates are installed, waptservice will stop and disable the WUA service until the next cycle.

State of Update on host

Windows updates can have 4 states on a host.

Status	Description
OK	A Windows update has installed correctly.
MISSING	A Windows update has not yet been downloaded to the WAPT server.
PENDING	**The WAPT Server knows it has to download** an update from official Microsoft servers.
DISCARDED	A Windows update was forbidden by rules.

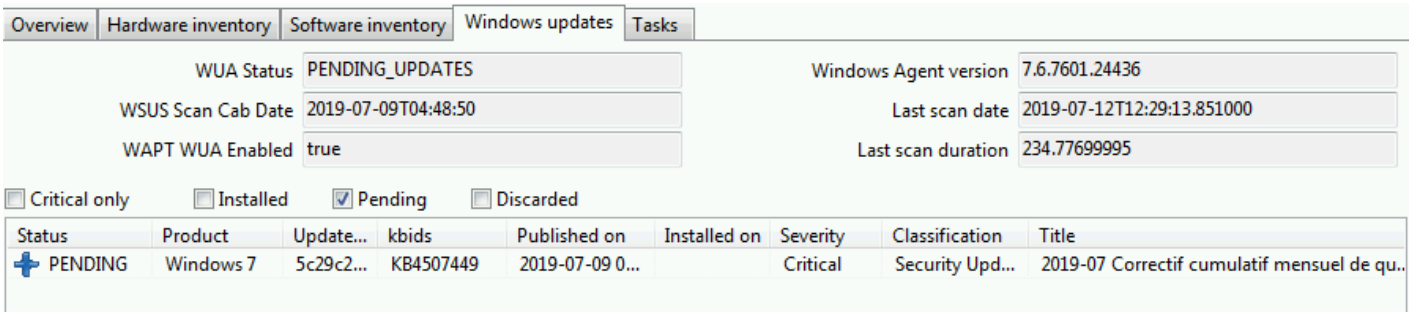


Fig. 10 – Pending Windows Updates showing in the WAPT console

Notion d'UpdateID

In WAPT we don't use kbids but instead we use **updateids**.

Cela nous permet d'être plus fin dans la gestion des mises à jour.

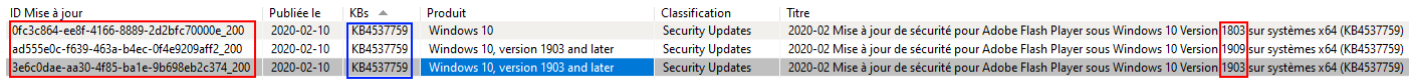


Fig. 11 – Duplicate kb

Dans cet exemple, la KB4537759 apparaît de multiples fois car il y a 3 différents updateids :

- win10 1803 ;
- win10 1903 ;
- win10 1909 ;

You should therefore authorize *updateids* and not *kb ids*.

31.3.8 WAPT does not force Windows to uninstall a 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.

If you really want to uninstall an update, you should package the KB that you want to uninstall as a standard WAPT package.

Voici un exemple :

```
from setuphelpers import *

uninstallkey = []

def install():
        with EnsureWUAServRunning():
                run('wusa /uninstall /KB:4023057')
```

31.3.9 Vidéo de démonstration

31.4 Using the reporting functions in WAPT

31.4.1 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.

With WAPT we offer to write your own SQL queries to display the result in the WAPT console, or to download already made queries from Internet.

31.4.2 Concepteur de requêtes WAPT

Le concepteur de requêtes vous offre la possibilité de modifier vos propres requetes sur la base de données PostgreSQL de WAPT.

Pour créer une nouvelle requête, cliquez sur *Reporting* → *Mode conception* → *Nouvelle requête*.

Indication :

- To rename a query, press the `F2` key.
- In the top banner, you can write your SQL query.

Pour éditer / modifier / Sauvegarder vos requêtes :

- The *Reload queries* button is used to reload queries saved on the server, for example, if a colleague has just edited a new query.
- The *New query* button will add a new blank query to the list.
- The *Delete query* button will delete the selected query from the WAPT Server.
- The *Export to Excel* button will export the result of your query to a spreadsheet.
- The *Save queries* button will save your query to the WAPT Server.
- The *Duplicate* button will duplicate an existing query to avoid writing a request from scratch.
- The *Execute* button executes the selected query.

Note :

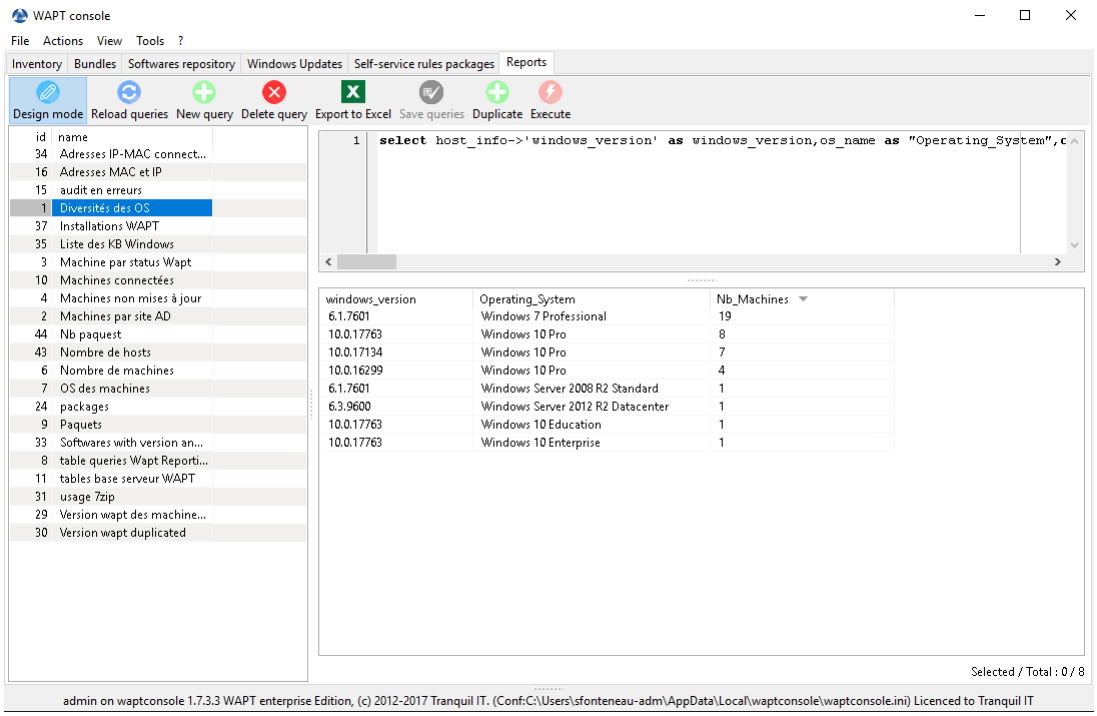


FIG. 12 – Concevoir une requête dans le reporting WAPT

— The queries are saved in the PostgreSQL WAPT database.
— The shortcut `CTRL+space` allows you to build your queries more effectively as it will auto-complete some fields for you.

31.4.3 Exemple de requêtes

Requêtes Ordinateur

— Counting hosts.

```
select count(*) as "Nb_Machines" from hosts
```

— Listing computers.

```
select
computer_name,
os_name,
os_version,
os_architecture,
serialnr
from hosts
order by 4,3,1
```

— Listing computers MAC addresses and IP.

```
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
```

— Listing Windows versions.

```
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
```

— Listing operating systems.

```
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
```

— Listing hosts not seen in a while.

```
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'
```

— Filtering hosts by chassis types.

```
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
```

— Listing of hosts with their Windows Serial Key.

```
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

— Listing WAPT packages in WAPT server repository.

```
select
package,
version,
architecture,
description,
section,
package_uuid,
count(*)
from packages
group by 1,2,3,4,5,6
```

— Listing hosts needing upgrade.

```
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

— Listing packages with their number of installation.

```
select
package,
version,
architecture,
description,
section,
package_uuid,
```

(suite sur la page suivante)

(suite de la page précédente)

```
count(*)
from hostpackagesstatus s
where section not in ('host','unit','group')
group by 1,2,3,4,5,6
```

Requête logiciel

— Listing WAPT Discovery agents.

```
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
s.key='WAPT_is1'
and (name ilike 'WAPT%%Discovery%%' or name ilike 'WAPT %%')
```

— Listing hosts with their 7zip version associated.

```
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
```

— Listing hosts with their software.

```
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
```

— Listing normalized software.

```
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 fais votre requête, idéalement avec une capture d'écran ou une table affichant un échantillon du résultat de votre requête.

31.4.4 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 standradiser le nom des logiciels avec WAPT.

— Click *Normalize Software Names* in the *Tools* menu.
— Select the software to standardize, for example, all different version of Adobe Flash Player.
— On the column *normalized*, press `F2` to assign a standardized name to the selected software. Then press `Enter`.

Note :
— To select several programs, select them with the `shift-up/down` key combination.
— You can also indicate a software like *windows update* or *banned* (Press `spacebar` in the corresponding column).

— Press on *Import* to load the changes from the server.
— Press on *Write* to save your changes.

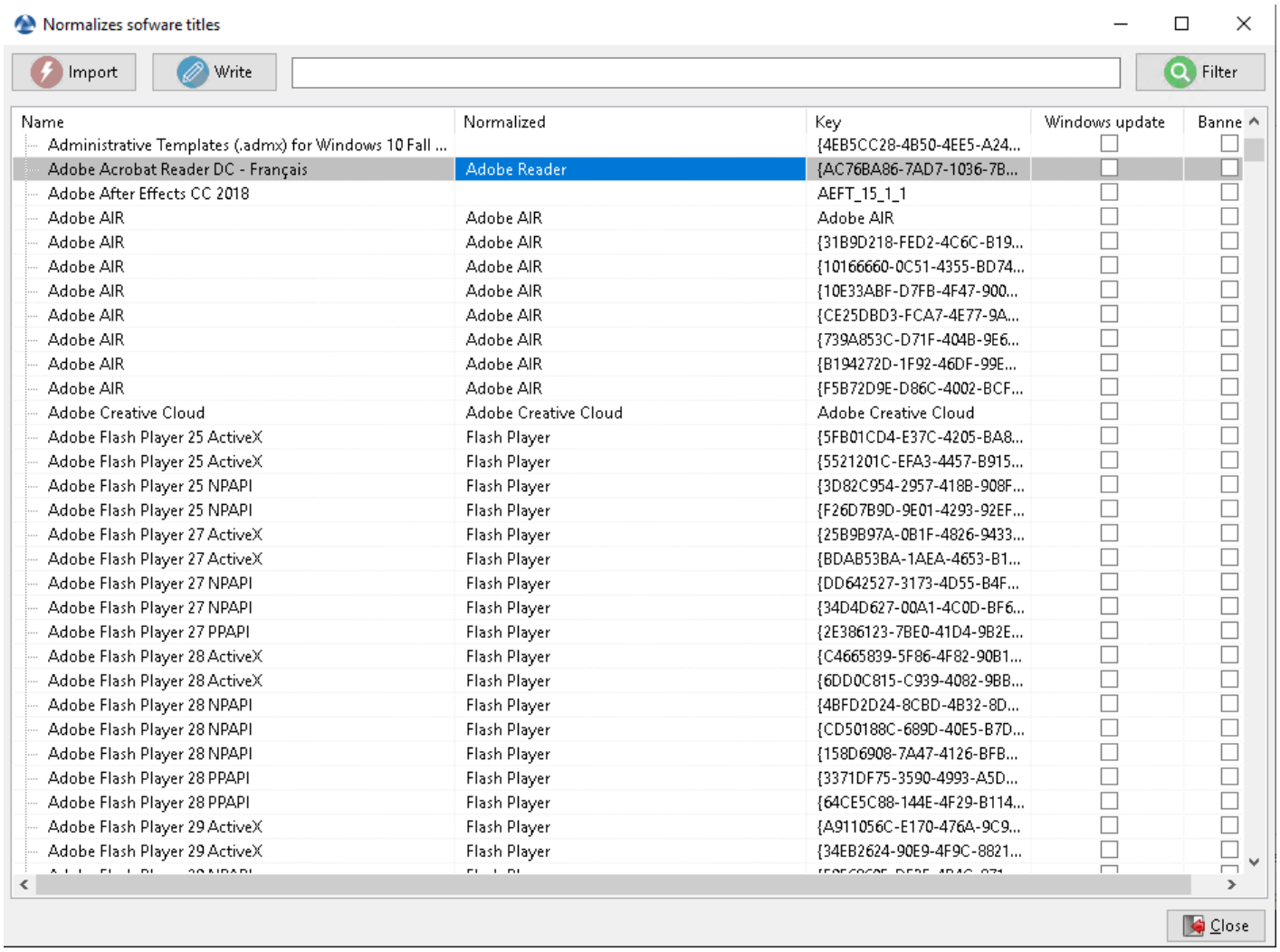
Vous pouvez maintenant lancer vos requêtes avec ce nom standardisé.

31.5 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.
— Find out in which version your PostgreSQL is.

```
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
```



Name	Normalized	Key	Windows update	Banne
Administrative Templates (.admx) for Windows 10 Fall ...		{4EB5CC28-4B50-4EE5-A24...	☐	☐
Adobe Acrobat Reader DC - Français	Adobe Reader	{AC76BA86-7AD7-1036-7B...	☐	☐
Adobe After Effects CC 2018		AEFT_15_1_1	☐	☐
Adobe AIR	Adobe AIR	Adobe AIR	☐	☐
Adobe AIR	Adobe AIR	{31B9D218-FED2-4C6C-B19...	☐	☐
Adobe AIR	Adobe AIR	{10166660-0C51-4355-BD74...	☐	☐
Adobe AIR	Adobe AIR	{10E33ABF-D7FB-4F47-900...	☐	☐
Adobe AIR	Adobe AIR	{CE25DBD3-FCA7-4E77-9A...	☐	☐
Adobe AIR	Adobe AIR	{739A853C-D71F-404B-9E6...	☐	☐
Adobe AIR	Adobe AIR	{B194272D-1F92-46DF-99E...	☐	☐
Adobe AIR	Adobe AIR	{F5B72D9E-D86C-4002-BCF...	☐	☐
Adobe Creative Cloud	Adobe Creative Cloud	Adobe Creative Cloud	☐	☐
Adobe Flash Player 25 ActiveX	Flash Player	{5FB01CD4-E37C-4205-BA8...	☐	☐
Adobe Flash Player 25 ActiveX	Flash Player	{5521201C-EFA3-4457-B915...	☐	☐
Adobe Flash Player 25 NPAPI	Flash Player	{3D82C954-2957-418B-908F...	☐	☐
Adobe Flash Player 25 NPAPI	Flash Player	{F26D7B9D-9E01-4293-92EF...	☐	☐
Adobe Flash Player 27 ActiveX	Flash Player	{25B9B97A-0B1F-4826-9433...	☐	☐
Adobe Flash Player 27 ActiveX	Flash Player	{BDAB53BA-1AEA-4653-B1...	☐	☐
Adobe Flash Player 27 NPAPI	Flash Player	{DD642527-3173-4D55-B4F...	☐	☐
Adobe Flash Player 27 NPAPI	Flash Player	{34D4D627-00A1-4C0D-BF6...	☐	☐
Adobe Flash Player 27 PPAPI	Flash Player	{2E386123-7BE0-41D4-9B2E...	☐	☐
Adobe Flash Player 28 ActiveX	Flash Player	{C4665839-5F86-4F82-90B1...	☐	☐
Adobe Flash Player 28 ActiveX	Flash Player	{6DD0C815-C939-4082-9BB...	☐	☐
Adobe Flash Player 28 NPAPI	Flash Player	{4BFD2D24-8CBD-4B32-8D...	☐	☐
Adobe Flash Player 28 NPAPI	Flash Player	{CD50188C-689D-40E5-B7D...	☐	☐
Adobe Flash Player 28 NPAPI	Flash Player	{158D6908-7A47-4126-BFB...	☐	☐
Adobe Flash Player 28 PPAPI	Flash Player	{3371DF75-3590-4993-A5D...	☐	☐
Adobe Flash Player 28 PPAPI	Flash Player	{64CE5C88-144E-4F29-B114...	☐	☐
Adobe Flash Player 29 ActiveX	Flash Player	{A911056C-E170-476A-9C9...	☐	☐
Adobe Flash Player 29 ActiveX	Flash Player	{34EB2624-90E9-4F9C-8821...	☐	☐

FIG. 13 – Normaliser le nom du logiciel

— Modify `pg_hba.conf` of the PostgreSQL version in use. In `/etc/postgresql/12/main/pg_hba.conf` for Debian and `/var/lib/pgsql/12/data/pg_hba.conf` for Centos under **# IPv4 local connections section**, add your address.

```
host    wapt            all             192.168.0.65/32             md5
```

où 192.168.0.65 est votre adresse IP autorisée à se connecter à la base de données WAPT.

— Allow PostgreSQL to listen on every interface in `/etc/postgresql/12/main/postgresql.conf` for Debian and `/var/lib/pgsql/12/data/postgresql.conf` for Centos, section **Connection Settings**.

```
listen_addresses = '*'
```

— Restart the service for your PostgreSQL version.

```
systemctl restart postgresql@12-main.service
```

— Connect to PostgreSQL on waptserver.

```
sudo -u postgres psql template1
```

— Then give a password to wapt user.

```
template1=# ALTER USER wapt WITH PASSWORD 'PASSWORD';
```

31.5.1 Vidéo de démonstration

31.6 Synchroniser les inventaires de WAPT vers GLPI

31.6.1 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.

31.6.2 Installer les dépendances requises

Pour pouvoir recevoir les inventaires sur votre seveur 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 *(. . ./glpi/plugins/fusioninventory/)*.

31.6.3 Configuration

You can open the window to configure Glpi with *Tools → Manage WAPT to Glpi*.

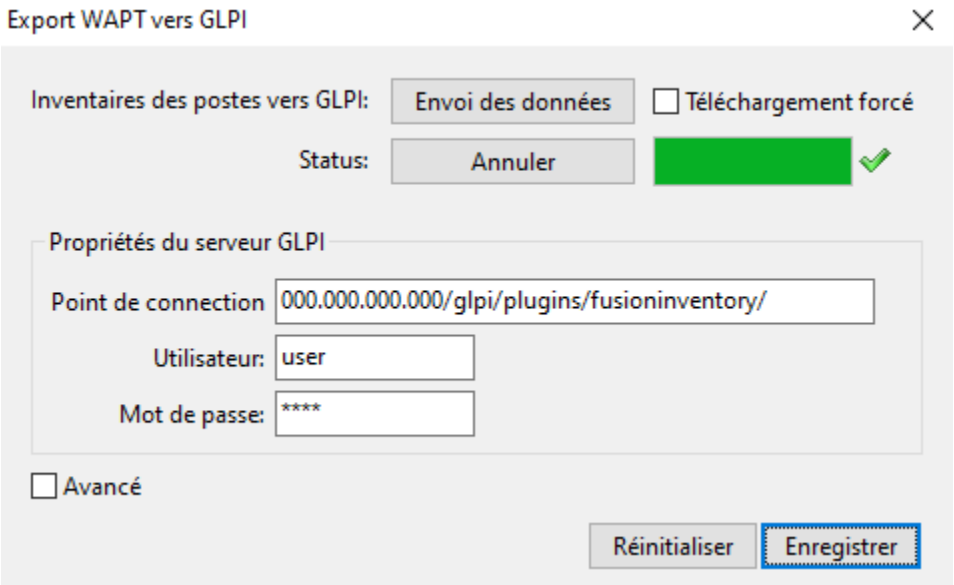


FIG. 14 – La console WAPT affichant la configuration de GLPI dans WAPT

In *Glpi Server Properties*, add the required parameters in the `waptserver.ini` configuration file.

```
[options]
...
glpi_server_endpoint = 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'invenaire 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 ou 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.

31.6.4 Utiliser WAPT pour envoyer les mises à jour d'invenaire 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 :

— When you fill in *Glpi Server Properties*, the configuration is registered on the WAPT server when you click *Save*.
— You can retrieve properties already registered on the WAPT server by clicking on *Reset* (the password is not loaded).
— The *Endpoint* field is the Glpi server url to send the inventories to the `glpi_server_endpoint`.
— You can trigger an upload without waiting for the scheduled task by clicking *Upload*.
— You can stop the upload at any time by clicking *Cancel*.

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 :

— With `Force upload` every inventory is uploaded, ignoring already uploaded data.

31.6.5 Advanced use of the Glpi plug-in

Pour afficher les propriétés avancées, sélectionnez le bouton *Avancé*.

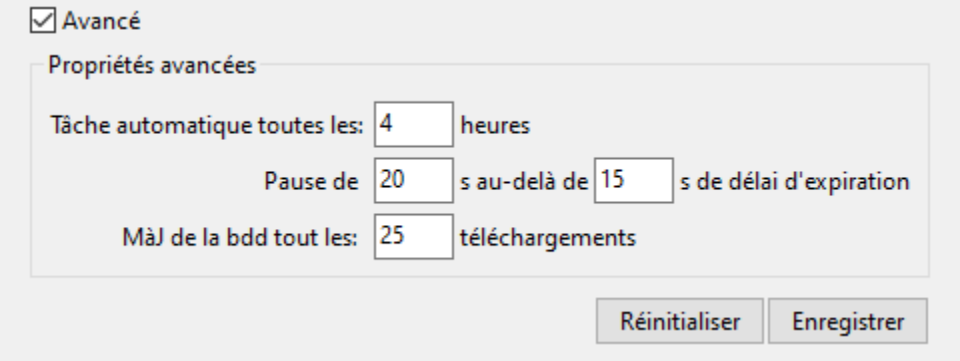


FIG. 15 – La console WAPT affichant la configuration avancée de GLPI dans WAPT

— La tâche planifiée tourne toutes les *Cron every…* heures seuelement 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 : If you want to disable automatic upload, you have to *Save* an empty *Endpoint*.

— You can trigger pauses (*Pause…*) when the server response time is too long (*over…*). ~> `glpi_server_pause_timeout`.
— *Update db…* sets the database synchronization frequency during upload. ~> `glpi_inventory_update_range`.

31.6.6 Ajouter des plugins dans la Console

To add custom plug-ins, go to *Tools* → *Preference* → *Plug-ins* Tab.

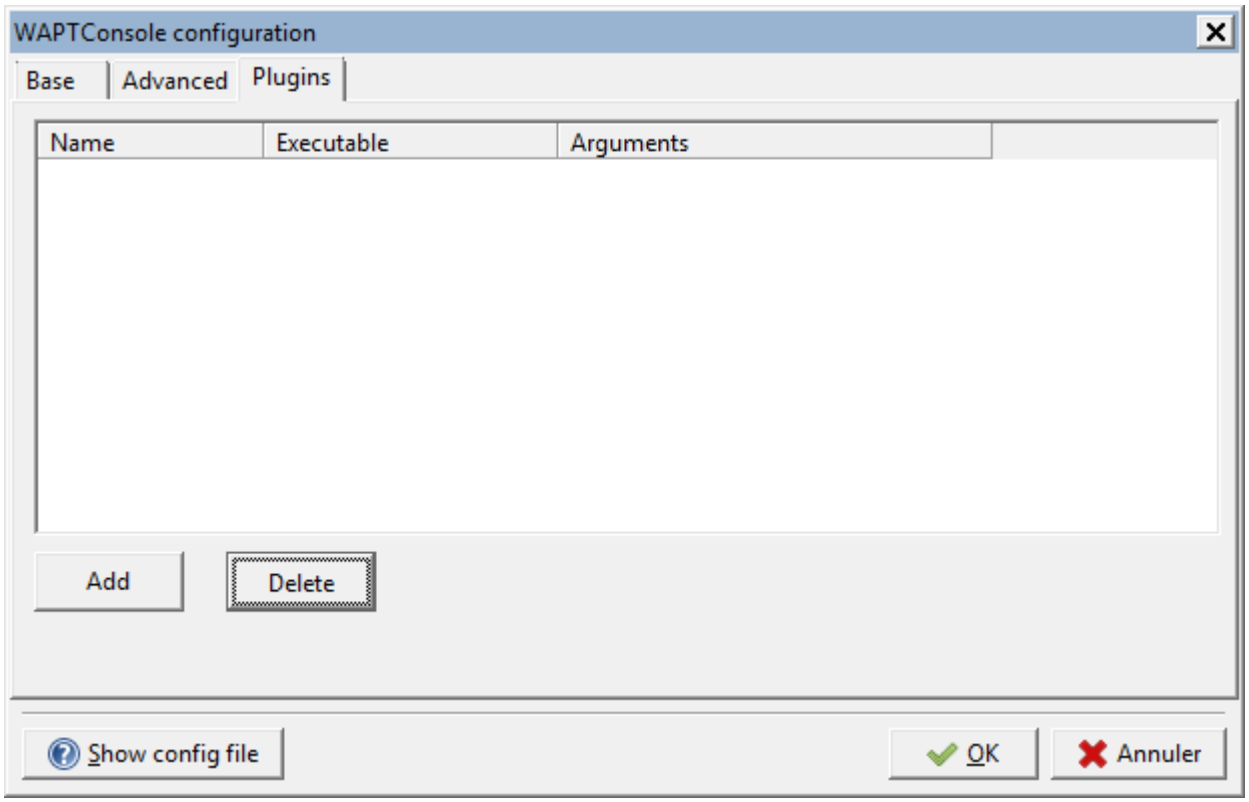


FIG. 16 – Creating a custom plug-in for WAPT

Click *Add* to add plug-ins, then edit the corresponding columns.

Colonne	Description
Nom	Name that will appear in the menu.
Exécutable	Path of the executable that will be executed after the click.
Arguments	**Arguments passed to the executable. Some variables can be used** like {ip}, {uuid} or {computer_fqdn}.

Plug-ins will then appear in the menu :

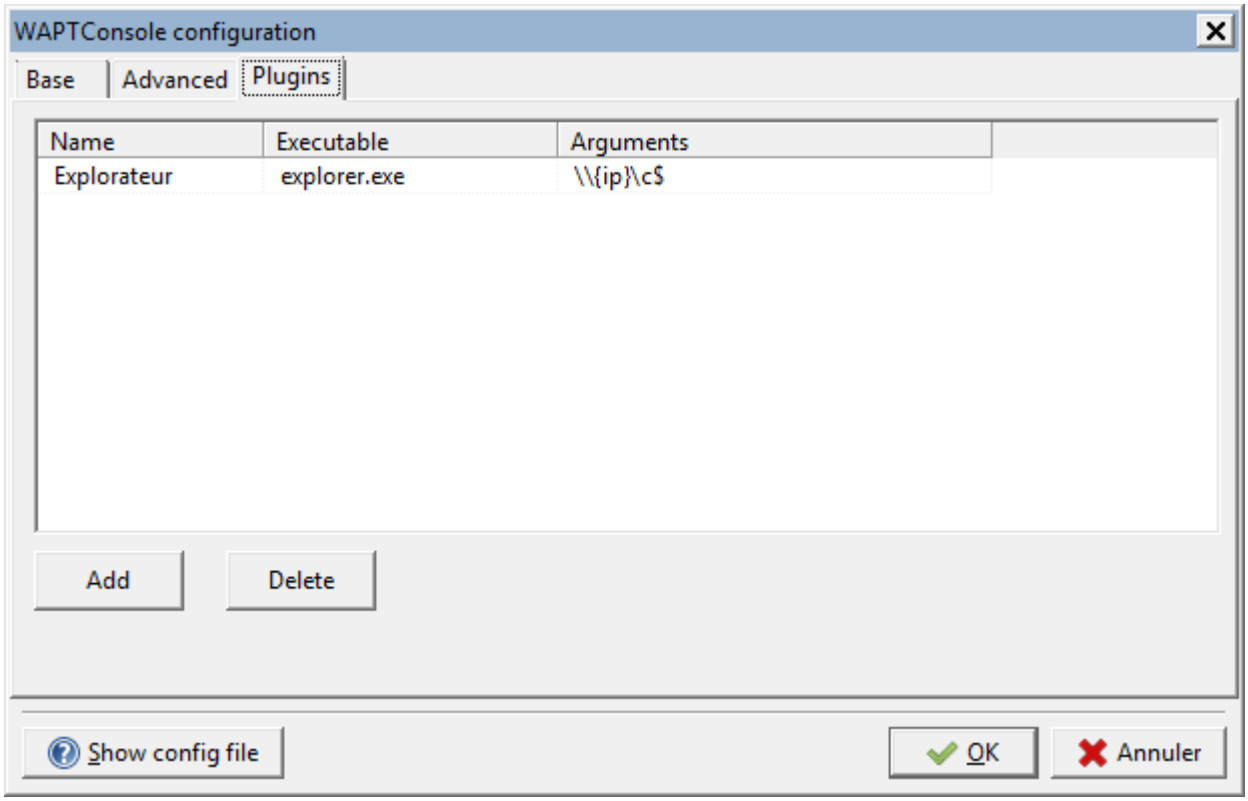


FIG. 17 – Insert « Explorer » as a plug-n with IP variables

31.6.7 L'état actuel de WAPT vers la passerelle GLPI

Les objets de l'inventaire qui sont actuellement téléversés vers GLPI par la passerelle WAPT-GLPI :

- Computer name / user name / description / OS name / OS version / language ;
- CPU / memory / battery / chassis type / physical or virtual ;
- Network card configuration ;
- Printer list and properties ;
- Installed software (not including system wide Appx install) ;
- Network drives ;
- **Environment variables. Note : currently both system and system-wide** user environment variables are included.

Objets de l'inventaire qui ne sont actuellement pas téléversé vers GLPI par la passerelle WAPT-GLPI :

- Display screens references ;
- Mouse and keyboard references ;
- Controllers card references (except graphic card) ;
- Antivirus version ;
- Firewall state ;
- Local group list ;
- Memory bank list and state ;
- USB ports list and connected devices ;
- Printer status ;
- Card readers ;
- System wide Appx list.

31.7 Re-signing all host packages from the WAPT console

This method for re-signing all host packages is useful when the underlying cryptographic method or library changes, as this is the case when upgrading from WAPT 1.8.2 (Python 2.7 based) and WAPT >= 2.0 (Python 3.x).

Indication : Use the Administrator's certificate for re-signing packages.

31.7.1 Les paquets hôtes

- Select all host.
- Right-click on the selected hosts.
- Select *Resign Host packages*.
- Confirm re-signing the selected hosts.
- Then, enter you private key password.
- Selected WAPT *host* packages are now all re-signed using the new cryptographic method required with Python3.

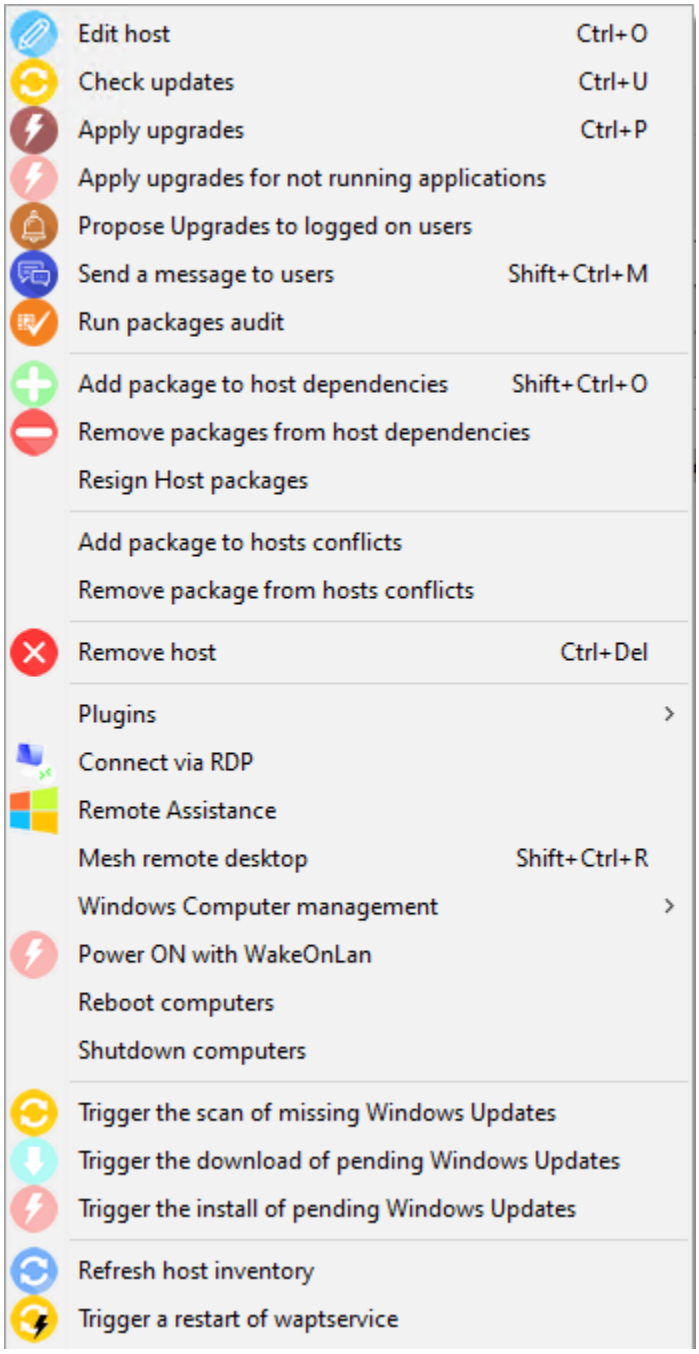


FIG. 18 – Le menu avec clic-droit

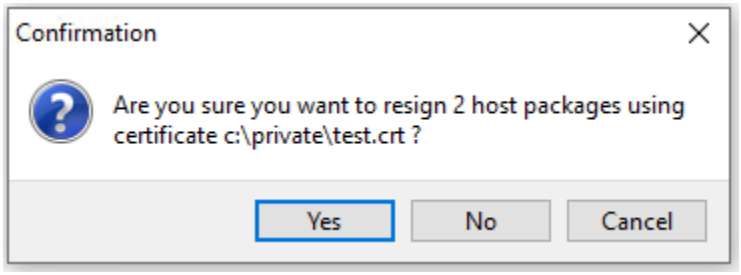


FIG. 19 – Confirmer la resignature des hôtes séléctionnés

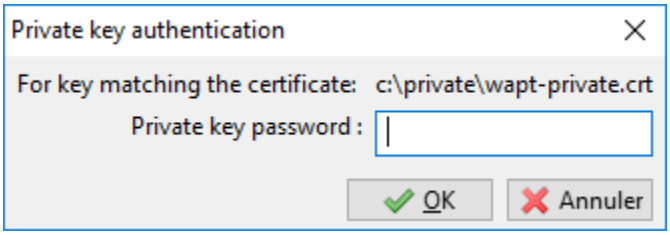


FIG. 20 – Entrer le mot de passe pour déchiffrer la clé privée

31.7.2 Autre type de paquet WAPT

— Open the repositories in your WAPT console.

— Select all packages in the repository, then right-click on the selection.

— Select *Resign packages*.
— To launch the signature process, click on *Resign packages*.
— After processing, which may take some time, all packages will have been re-signed.

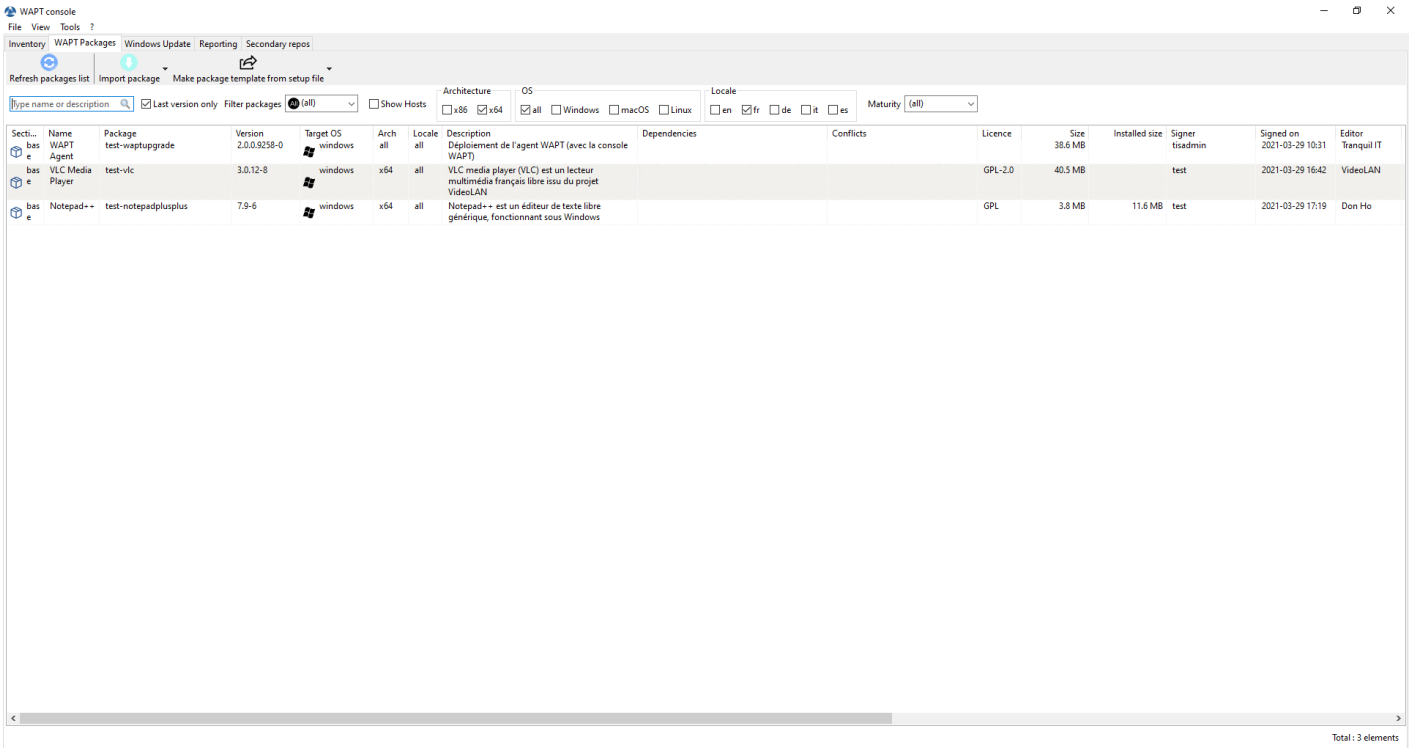


FIG. 21 – Les dépôts disponibles sur la console WAPT

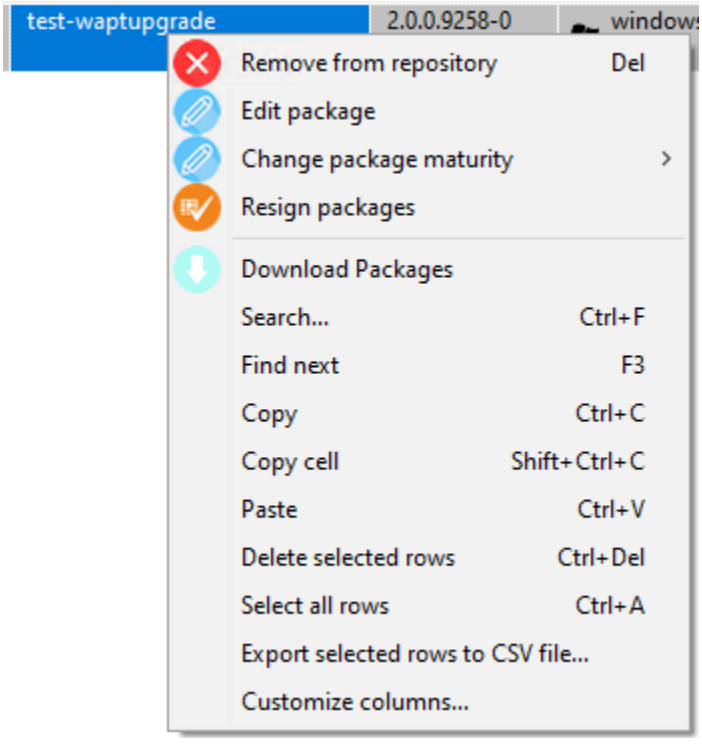


FIG. 22 – Le menu avec clic-droit

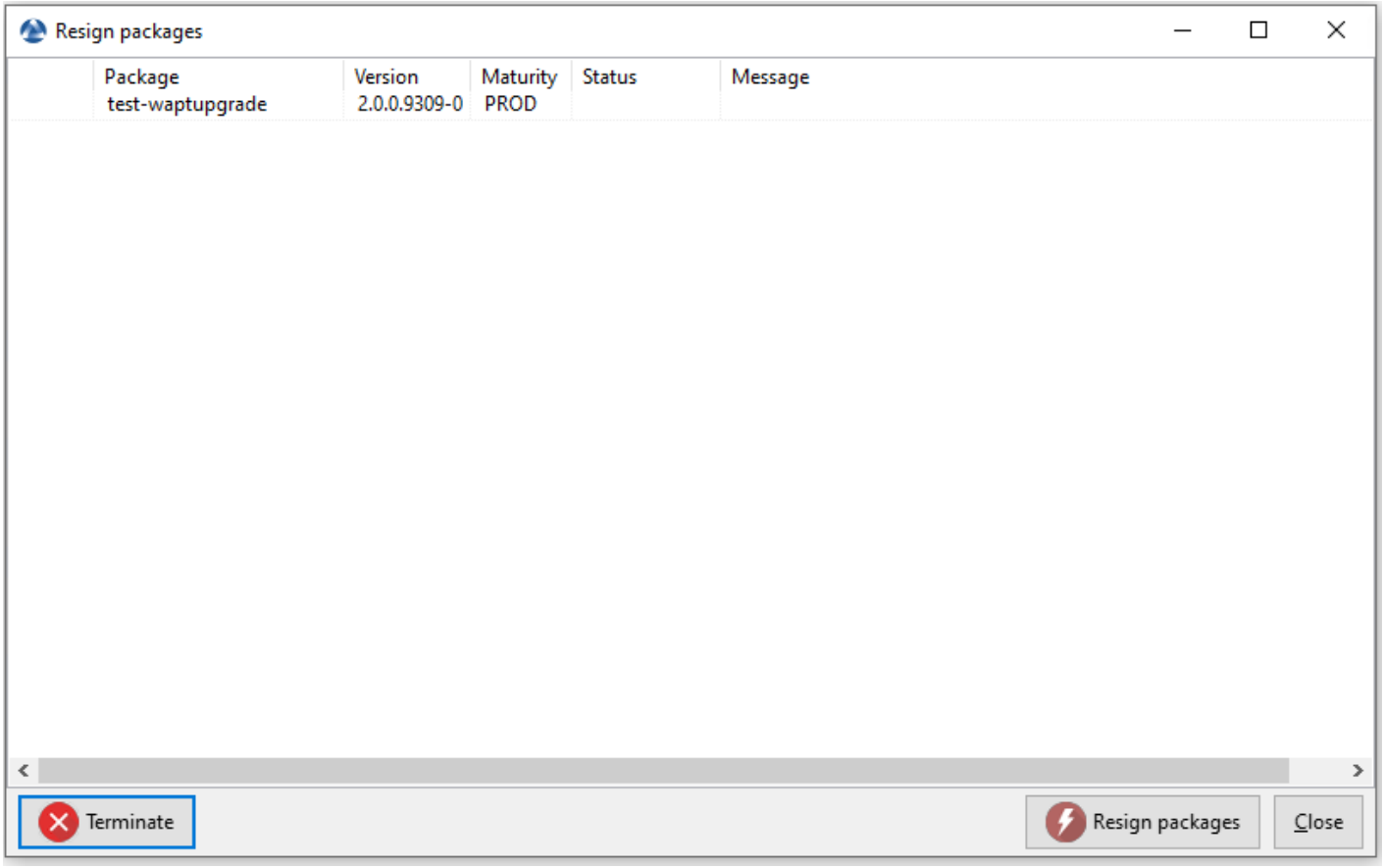


FIG. 23 – Fenêtre de re-signature des paquets WAPT

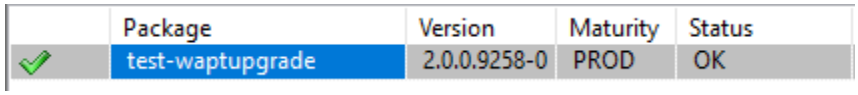


FIG. 24 – Signature processing has ended successfully

CHAPITRE 32

Utiliser le WAPT Self-Service

32.1 Présentation

Avec WAPT 1.7 **Enterprise** vous pouvez désormais filter la liste des paquets self-service disponibles pour vos utilisateurs.

Vos utilisateurs pourront installer une sélection de paquets WAPT sans être un *Administrateur Local* sur leur poste.

Les *Utilisateurs* gagnent en autonomie en déployant des logiciels et des configurations qui sont de confiance et autorisées par l'*Organisation*. C'est une fonctionnalité avec un gain de temps pour le support informatique utilisateur de l'Organisation.

32.2 Comment cela fonctionne ?

Avec WAPT 1.7 **Enterprise**, un nouveau type de paquet WAPT est apparu en plus des paquets *base*, *group*, *host*, *profile* et *unit* : ce sont les paquets **self-service**.

un paquet *self-service* peut être déployé sur les hôtes pour lister les différentes règles self-service à appliquer sur l'hôte.

32.3 Comment utiliser la fonctionnalité self-service ?

Indication : La fonctionnalité **self-service** n'est disponible qu'en WAPT **Enterprise**.

Dans la version **Discovery**, seuls les Administrateurs locaux et les membres du group *waptself-service* peuvent accéder au self-service de l'agent.

Dans la version **Discovery**, il n'est pas possible de filtrer les paquets accessibles ou non aux utilisateurs.

Dans la console, allez dans l'Onglet de règles *Self-service*.

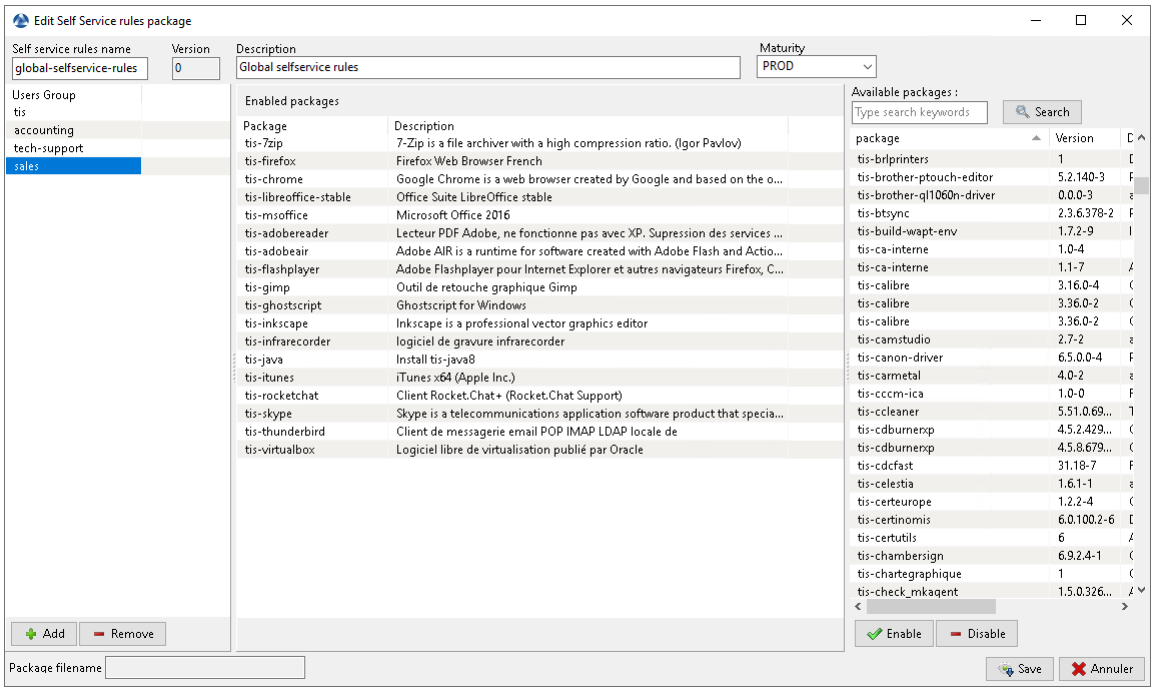


FIG. 1 – Créer un paquet *self-service*

Vous pouvez désormais créer votre premier paquet de règle *self-service*.

- donnez un nom à votre nouveau paquet *self-service* ;
- cliquez sur *Ajouter* pour ajouter un groupe Active Directory (en bas à gauche) ;
- nommez le groupe *self-service* (avec `F2` ou tapez directement dans lacellule) ;
- glissez les paquets logiciels et de configuration autorisés pour ce groupe *self-service* vers la colonne centrale ;
- ajoutez autant d egroupes que vous le souhaitez dans le paquet ;
- sauvegardez le paquet et déployez le paquet sur votre sélection d'hôtes ;
- une fois le paquet déployé, seuls les paquets autorisés listés dans les groupes *self-service* donc l'*Utilisateur* est membre seront affichés à l'*Utilisateur* conncecté ;

Note :

- si un groupe apparaît dans plusieurs paquets *self-service*, alors les règles sont fusionnées ;
- l'authentification utilisé est celui du système, les utilisateurs locaux et les groupes mais si la machine est dans un domaine alors l'authentification et les gorupes vont aussi fonctionner avec les utilisateurs et les groupes du domaine ;

32.4 Comment utiliser le self-service sur un poste utilisateur ?

Le self-service est accessible au utilisateurs dans le menu démarrer sous le nom de *Self-Service logiciels WAPT*.

Il est aussi disponible directement dans `<base>\waptself.exe`.

Li'dentifiant 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 peur avoir plus de détails sur chaque paquet avec l'icône + ;

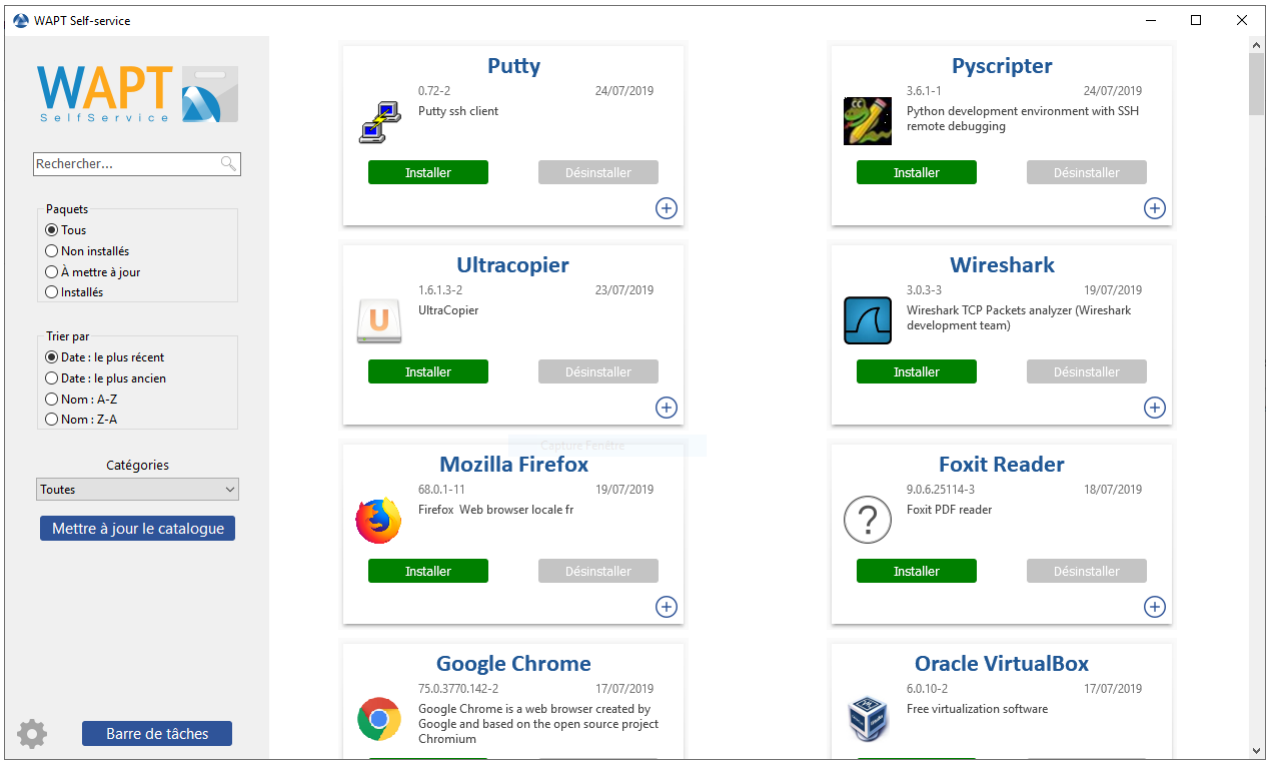


Fig. 2 – Self Service

— différents filtres sont disponibles pour l'utilisateur sur le panneau de gauche ;
— le bouton *Mettre à jour le catalogue » est utilisé pour forcer un :command : `wapt-get update* sur l'agent WAPT ;
— la list des catégories de paquet est affiché à l'utilisateur. Pour ajouter une catégorie à la liste, vous devez spécifier la catégorie dans la section *categories* du fichier `control` du paquet concerné ;
— 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 bouton *configuration* en bas à gauche.

32.5 Personnaliser l'interface du Self Service

32.5.1 Ajouter un le Logo de votre Organisation

Dans la **version Enterprise de WAPT uniquement**, il est possible de changer le logo qui apparaît dans l'interface du self-service et ainsi améliorer l'acceptation du Self Service par vos utilisateurs.

Pour ce faire, placez simplement le logo que vous voulez dans `<wapt>\templates\waptself-logo.png`

Note : Il est fortement recommandé d'utiliser un fichier *.png* avec une rsolution de *200 x 150px*.

32.5.2 Gérer les catégories de paquet

Les catégories par défaut sont :
- Internet ;
- Utilitaires ;
- Messagerie ;
- Securite ;
- Syteme et reseau ;
- Stockage ;
- Media ;
- Developpement ;
- Bureautique ;

Vous pouvez créer vos propres catégories facilement en remplissant la section `categories` du fichier `control` de n'importe quel paquet WAPT et écrire une nouvelle catégorie de votre choix, WAPT va automatiquement montrer le paquet dans la nouvelle catégorie.

32.6 Les configurations de l'agent WAPT pour le WAPT Self-Service

L'agent WAPT peut être configuré pour forcer les paquets WAPT self-service à être filtrés pour les Administrateurs Locaux *Paramètres d'authentification de Self-Service WAPT et Waptservice*.

32.7 Configurer une méthode d'authentification différente pour le self-service

Comme mentionné ci-dessus, l'authentification sur le service WAPT est configuré par défaut sur le mode système.

Cela signifie que le service WAPT transmet l'authentification directement au système d'exploitation ; il récupère également les groupes directement en interrogeant le système d'exploitation.

Ce comportement est défini avec la valeur du `service_auth_type` dans `wapt-get.ini`. » La valeur par défaut est *system*.

dans ce mode, nous supposons que les Administrateurs locaux peuvent voir tous les paquets. pour changer ce comportement, modifiez la valeur de `waptservice_admin_filter` dans `wapt-get.ini`.

Il se pourrait que vous soyez intéressé par cet article décrivant les *configurations pour WAPT Self-Service et Waptservice Authentification* pour plus d'options.

Deux modes additionnels sont disponibles :
- `waptserver-ldap` : ce mode permet une authentification avec le serveur WAPT.Le serveur WAPT fera une requête LDAP pour vérifier l'authentification et les groupes. **Attention** ! Pour que cela puisse marcher, vous devez avoir configuré l'authentification LDAP du serveur WAPT, (la configuration du groupesera ignoré ») Voir *cet article pour configurer l'authentication avec Active Directory* pour plus d'information. »
- `waptagent-ldap`, ce mode permet une authentification avec le serveur LDAP identifié dans le `wapt-get.ini`. L'agent WAPT va faire une requête LDAP pour vérifier l'authentification et les groupes.
  Il se pourrait que vous soyez intéressé par cet article décrivant les *configurations pour WAPT Self-Service et Waptservice Authentification* 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.

32.8 Vidéo de démonstration

CHAPITRE 33

Utiliser le WAPTtray

wapttray est un utilitaire fonctionnant en contexte utilisateur, il est situé dans le dossier WAPT `C:\Program Files (x86)\wapt`.

wapttray s'ouvre à l'ouverture de session si l'option a été cochée pendant l'installation. L'icône apparaîtra dans la barre d'outils Windows.

Nous pouvons aussi lancer manuellement le **wapttray** avec une GPO au démarrage pointant sur `C:\Program Files (x86)\wapt\wapttray.exe`.

L'icône est utile pour des utilisateurs autonomes qui veulent choisir le bon moment pour mettre à jour leur paquet.

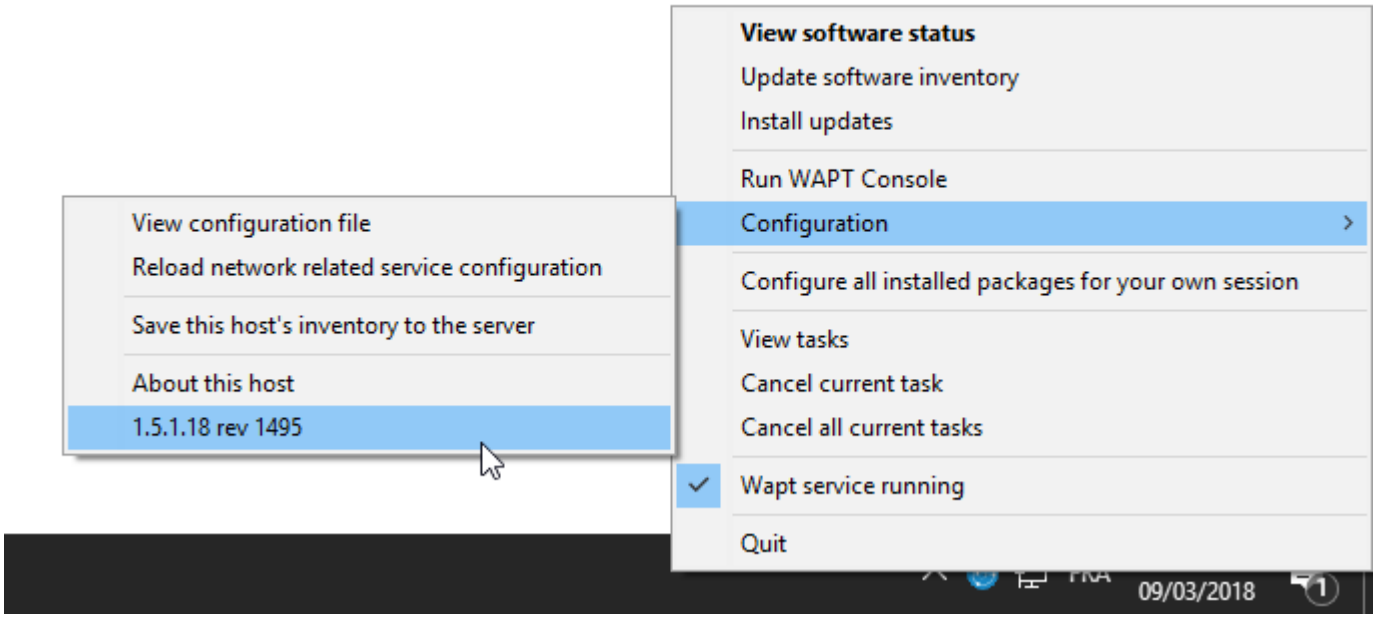


FIG. 1 – WAPTtray dans la barre de notification Windows

33.1 Les fonctionnalités du WAPTtray

TABLEAU 1 – Liste des fonctionnalités du WAPTtray

Action	Description
Afficher le statut des logiciels	lance l'interface web local dan sun navigateur
Installer les mises à jour	lance l'installation d'une mise à jour en attente
mettre à jour l'inventaire des logiciels	rafraîchir la liste de paquet disponibles. Double-clic sur l'icone fait la même action.
Lancer la console WAPT	lance la console WAPT
Afficher le fichier de configuration	ouvre le fichier `C:\Program Files (x86)\wapt\wapt-get.ini` avec des privilèges d'*Administrateur Local* (des identifiants peuvent être demandés)
Recharger la configuration réseau du service	recharge la connexion au serveur WAPT dans le cas d'une reconfiguration réseau
Sauvegarder l'inventaire de cette machine sur le serveur	met à jour l'inventaire du poste sur le serveur WAPT
Configurer tous les paquets installés pour votre session	lance un **`session-setup`** pour configurer tous les paquets installés dans l'environnement utilisateur
Annuler toutes les tâches en cours	montre les tâches en cours, permet d'annuler une tâche en cours, permet d'annuler toutes les tâches en cours
Arrêter et démarrer le service WAPT	arrête et relance le *WAPTservice*
Quitter	ferme l'"*icone* sans stopper le *WAPTservice* local

CHAPITRE 34

Utiliser le WAPTExit

waptexit vous permet de mettre à jour et d'installer des paquets WAPT lorsqu'un poste s'éteint, à la demande de l'utilisateur, ou à un moment défini.

Le mécanisme est simple. Si les paquets sont en attente de mise à jour, ils seront installés.

Indication : Quand utiliser WAPTexit ?

La méthode du WAPTexit est très efficace dans la plupart des situations car il n'a pas besin d'une intervention de l'"*Utilisateur* ou de l'"*Administrateur*.

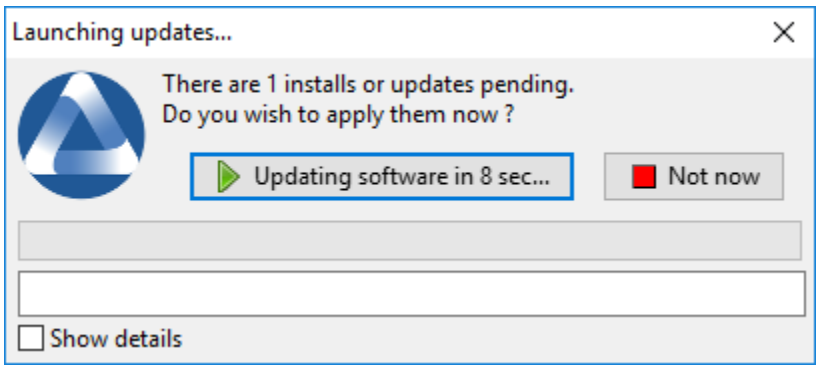


FIG. 1 – Fenêtre WAPTexit
WAPTexit

waptexit s'exécute par défaut à l'extinction, il est installé par défaut avec l'agent WAPT.

Le comportement du **waptexit** est personnalisable dans `C:\Program Files (x86)\wapt\wapt-get.ini`.

34.1 Déclencher manuellement l'exécution du WAPTexit

En créant un raccourci sur le bureau, on peut permettre aux utilisateurs de lancer les mises à jour par eux-même au moment qui leur convient en cliquant simplement sur l'icône *WAPTexit*.

Le comportement du **waptexit** est personnalisable dans `C:\Program Files (x86)\wapt\wapt-get.ini`.

34.2 Déclencher le WAPTexit avec une tâche planifiée

On peut déployer une GPO ou un paquet WAPT qui va déclencher le WAPTexit à un moment pré-défini.

Déclencher le WAPTexit avec une tâche planifiée est ce qu'il y a de plus convenable pour les serveurs qui ne s'éteignent pas fréquemment.

Vous pouvez adapter la procédure en décrivant comment l'agent WAPT va *déclencher le script WAPTexit.exe au moment de votre choix*.

Indication : vous pouvez utiliser le script suivant pour votre tâche planifiée, adaptée à votre besoin (**Enterprise uniquement**) :

```
waptpython -c "from waptenterprise.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 qui sont mis à jour seront coupé avec une potentielle perte de données. WAPTexit peut échouer la mise à jour d'un logiciel si ce dernier est dans la liste `impacted_process` du fichier `control` ou un des logiciels que vous tentez de mettre à jour. Voir *ci-dessous* pour plus d'information. »

La méthode pour déclencher le WAPTexit à un moment planifié est la mthode la moins recommendée pour les postes. Il est préférable de laisser le WAPTexit s'exéctuer à l'extinction ou à la demande de l'utilisateur.

34.3 Empêcher l'annulation des mises à jour

Pour désactiver l'interruption de l'installation des msies à jour vous pouvez lancer le **waptexit** avec l'argument :

```
waptexit.exe -allow_cancel_upgrade = True
```

Autrement **waptexit** va prendre la valeur indiquée dans le `C:\Program Files (x86)\wapt\wapt-get.ini` :

```
[global]
allow_cancel_upgrade = False
```

Si cette valeur n'est pas indiquée dans `C:\Program Files (x86)\wapt\wapt\wapt-get.ini`, alors la valeur par défaut sera **10**.

34.4 Augmenter le temps de déclenchement dans waptexit

Pour spécifier un temps d'attente avant le démarrage automatique des installations vous pouvez lancer **waptexit** avec l'argument :

```
waptexit.exe -waptexit_countdown = 10000
```

Autrement **waptexit** va prendre la valeur indiquée dans le `C:\Program Files (x86)\wapt\wapt-get.ini` :

```
[global]
waptexit_countdown = 25
```

Si cette valeur n'est pas indiquée dans `C:\Program Files (x86)\wapt\wapt\wapt-get.ini`, alors la valeur par défaut sera **1**.

34.5 Ne pas interrompre l'activité de l'utilisateur

Pour dire à WAPT de ne pas lancer un **upgrade** des logiciels sur la machine (attribut *impacted_process* dans le paquet), vous pouvez lancer **waptexit** avec l'argument :

```
waptexit.exe -only_if_not_process_running=True
```

Autrement **waptexit** va prendre la valeur indiquée dans le `C:\Program Files (x86)\wapt\wapt-get.ini` :

```
[global]
upgrade_only_if_not_process_running = True
```

Si cette valeur n'est pas indiquée dans `C:\Program Files (x86)\wapt\wapt\wapt-get.ini`, alors la valeur par défaut sera **False**.

34.6 Lancer l'installation des paquets avec un niveau de priorité spécifique

Pour dire à WAPT de ne mettre à jour que les paquets prioritaires, vous pouvez lancer le **waptexit** avec l'argument :

```
waptexit.exe -priorities = high
```

Autrement **waptexit** va prendre la valeur indiquée dans le `C:\Program Files (x86)\wapt\wapt-get.ini` :

```
[global]
upgrade_priorities = high
```

Si cette valeur n'est pas indiquée dans `C:\Program Files (x86)\wapt\wapt\wapt-get.ini`, alors la valeur par défaut sera **Vide** (pas de filtre sur la priorité).

34.7 Personnaliser le WAPTexit

Il est possible de personnaliser le waptexit en plaçant l'image que vous voulez dans `C:\Program Files (x86)\wapt\templates\waptexit-logo.png`.

34.8 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
```

CHAPITRE 35

Utiliser WAPT en ligne de commande

L'agent WAPT installé permet l'utilisation de WAPT en ligne de commande à l'aide du programme **`wapt-get`**.

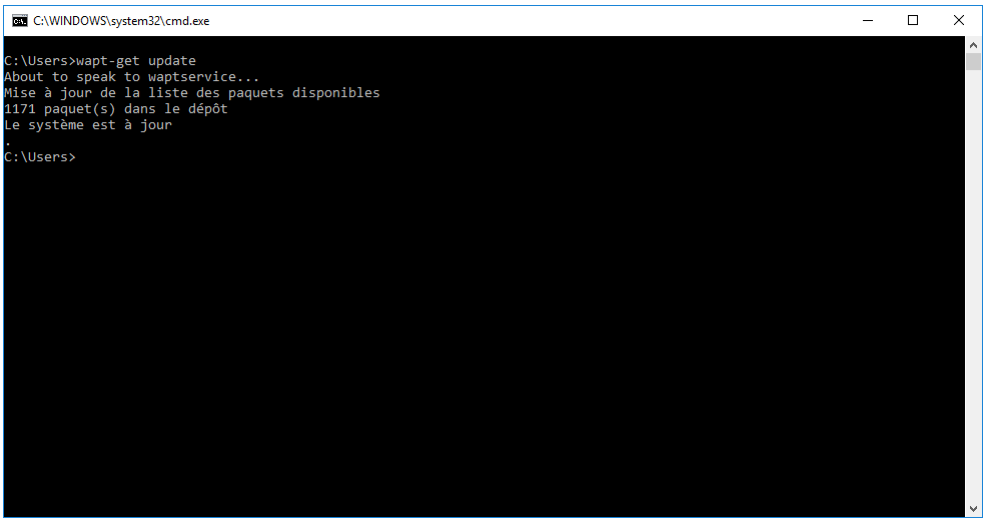


FIG. 1 – Invite de commande Windows

Note :

- By default, command-line actions in WAPT are executed with the rights of the user who launched the **`cmd.exe`**.
- If the *User* is not a *Local Administrator* or if the **`cmd.exe`** has not been launched with *Local Administrator* privileges, the command will be passed on to the **`waptservice`**.
- For security reasons, some actions will require a login and a password.
- Only *Local Administrators* and members of the *waptselfservice* Active Directory security group are allowed.
- To force using the WAPT service as a *Local Administrator*, simply add *-S* after **`wapt-get.exe`**.

35.1 Utiliser les fonctions les plus courantes de la ligne de commande

35.1.1 wapt-get update

La commande **update** permet de mettre à jour la liste des paquets disponibles.

L'agent WAPT local télécharge le fichier `Packages` depuis le dépôt et le compare avec sa base de données.

— If new updates are available, the WAPT agent switches the packages status to **TO-UPGRADE**.
— If new software have been added on the repository, they become downloadable by the WAPT agent.

Note : L'action **update** ne télécharge pas les paquets, elle met uniquement à jour la base locale des paquets.

La commande `wapt-get update` renvoie :

```
Update package list
Total packages: 751
Added packages:

Removed packages:

Upgradable packages:
upgrade
additional
install
remove
Repositories URL:
https://srvwapt.mydomain.lan/wapt
https://srvwapt.mydomain.lan/wapt-host
```

35.1.2 wapt-get upgrade

La commande **upgrade** lance l'installation des paquets en attente de mise à jour ou d'installation.

L'agent WAPT local télécharge si nécessaire les paquets WAPT dans son cache local puis installe les paquets.

Indication : Il est recommandé de lancer un **update** avant de lancer un **upgrade** ;

Sans un **update** préalable, l'agent WAPT n'installera rien ;

La commande `wapt-get upgrade` renvoie :

```
Installing tis-mumble
Shutting down Mumble
installing Mumble 1.2.8
Installing w7demo.domain.lan

=== install packages ===
w7demo.domain.lan (=3) | w7demo.domain.lan (3)
```

(suite sur la page suivante)

(suite de la page précédente)

```
=== additional packages ===
tis-mumble                    | tis-mumble (1.2.8-1)
```

35.1.3 wapt-get search

La commande **search** permet de rechercher un ou des paquet(s) sur les dépôt enregistrés.

La recherche prend un argument en paramètre. Cet argument servira à chercher dans le nom du paquet et dans sa description.

La commande `wapt-get install tis-firefox` renvoie :

Nom du paquet	Version	Plateforme	Description
tis-firefox	50.0.2-73	all	Mozilla Firefox Web Browser in French
tis-firefox-en	50.0.1-58	all	Mozilla Firefox Web Browser in English
tis-firefox-esr	45.6.0-4	all	Mozilla Firefox Web Browser ESR
tis-flashplayer	24.0.0.186-1	all	Adobe Flashplayer for Firefox

35.1.4 wapt-get install

La commande **install** lance l'installation d'un paquet.

La commande prend un argument en paramètre. Cet argument est le nom du paquet contenant le préfixe.

Pour l'installation de Mozilla Firefox il faudra taper la commande `wapt-get install <prefix>-firefox`.

Note : Si le paquet n'est pas téléchargé au préalable, la commande **install** téléchargera préalablement le paquet en cache.

Attention : L'installation d'un paquet WAPT avec **install** n'ajoute pas le paquet en dépendance au poste.

Il est installé sur la machine, mais en cas de réinstallation du poste il ne sera pas réinstallé automatiquement.

La commande `wapt-get install tis-firefox` renvoie :

```
installing WAPT packages tis-firefox
Installing tis-firefox.local/wapt/tis-firefox_50.0.2-73_all.wapt: 44796043 / 44796043 (100%) (33651␣
↪KB/s)
Firefox Setup 50.0.2.exe successfully installed.
Disabling auto update
Disabling profile migration from ie
Override User UI

=== install packages ===
tis-firefox                   | tis-firefox (50.0.2-73)
```

35.1.5 wapt-get remove

La commande **remove** lance la désinstallation d'un paquet.

La commande prend un argument en paramètre. Cet argument est le nom du paquet contenant le préfixe.

Pour désinstaller Mozilla Firefox il faudra taper la commande : **wapt-get remove <prefix>-firefox**.

Attention : La désinstallation d'un paquet WAPT avec **remove** ne supprime pas la dépendance entre le paquet machine et le paquet.

Le paquet sera effectivement désinstallé de la machine, mais il sera réinstallé automatiquement lors du prochain **upgrade**.

Afin de supprimer totalement un paquet d'un poste il faut faire un **remove** puis éditer la configuration de la machine via la console et supprimer la dépendance au paquet.

La commande `wapt-get remove tis-firefox` renvoie :

```
Removing tis-firefox ...

=== Removed packages ===
  tis-firefox
```

35.1.6 wapt-get clean

La commande **clean** supprime du cache les paquets dans le répertoire `C:\Program Files (x86)\wapt\cache`.

La commande **clean** est lancée à chaque fin de mise à jour pour économiser le stockage sur le disque dur de la machine.

La commande `wapt-get clean` renvoie :

```
Removed files:
C:\Program Files (x86)\wapt\cache\tis-mumble_1.2.8-1_all.wapt
C:\Program Files (x86)\\wapt\cache\tis-vlc_2.2.4-2_all.wapt
```

35.2 Utiliser les lignes de commande spéciales

35.2.1 wapt-get register

La commande `wapt-get register <description>` permet de remonter l'inventaire matériel et logiciel de la machine sur le serveur d'inventaire WAPT.

Indication : Vous pouvez préciser une description supplémentaire en paramètre de **register**, cette description sera affichée dans la console WAPT comme description du poste.

Vous pouvez donc profiter de WAPT pour améliorer la gestion administrative de votre parc en affectant à la machine le nom d'un service ou le nom du principal utilisateur de la machine.

The command **wapt-get register "John Doe PC** returns nothing.

35.2.2 wapt-get download

La commande `wapt-get download <nom du paquet>` télécharge le paquet dans le cache local WAPT : `C:\Program Files\wapt\cache`.

La commande `wapt-get download tis-7zip` renvoie :

```
Downloading packages tis-7zip (=16.4-8)

Downloaded packages:
  C:\Program Files (x86)\wapt\cache\tis-7zip_16.4-8_all_all.wapt
```

35.2.3 wapt-get download-upgrade

La commande `wapt-get download-upgrade` télécharge les paquets à mettre à jour, en prévision de leur installation, dans le cache local WAPT `C:\Program Files (x86)\wapt\cache`.

La commande `wapt-get download-upgrade` renvoie :

```
=== downloaded packages ===
C:\Program Files (x86)\wapt\cache\tis-firebird_2.5.5.26952-1_all.wapt
```

35.2.4 wapt-get show

La commande `wapt-get show <nom du paquet>` affiche les informations contenues dans l'index `Packages`.

Si plusieurs versions du paquet sont disponibles sur le dépôt, toutes les versions sont affichées.

La commande `wapt-get show tis-firebird` renvoie :

```
Display package control data for tis-firebird

package           : tis-firebird
version           : 2.5.5.26952-1
architecture      : all
section           : base
priority          : optional
maintainer        : Hubert TOUVET
description       : Firebird database SQL superserver with admin tools (Firebird Project)
filename          : tis-firebird_2.5.5.26952-1_all.wapt
size              : 7012970
repo_url          : https://srvwapt.mydomain.lan/wapt
md5sum            : 6f6d70630674f5d58a5259b1e6752221
repo              : global
```

35.2.5 wapt-get list

La commande `wapt-get list` affiche la liste des paquets WAPT actuellement installés sur la machine.

La commande `wapt-get list` renvoie :

package	version	install status	install_date	description
tis-7zip	16.4-8	OK	2016-12-01T17 :43	compression archivage 7-zip pour x86 et x64
tis-brackets	1.8-1	OK	2016-12-01T17 :44	Brackets propose un éditeur de code léger, libre et développé par Adobe.
tis-ccleaner	5.23.5808-0	OK	2016-12-01T18 :55	utilitaire de choix pour nettoyer, réparer et optimiser rapidement Windows
tis-rsat-win7x64	2	OK	2016-12-02T10 :46	package for MS RSAT Remote server admin windows6.1-kb958830-x64 pour Win7 SP1
tis-rsat-x64	1	OK	2016-12-02T10 :51	package for MS RSAT Remote server admin windows6.1-kb958830-x64 pour Win7 SP1
tis-dotnetfx4.6	4.6.2-1	OK	2016-12-09T16 :05	dot net FX 4.6.2 Framework CLient. replace 4/4.5/4.5.1/4.5.2/4.6/4.6.1

35.2.6 wapt-get upgradedb

La commande `wapt-get upgradedb` met à jour le schéma de la base de données locale WAPT si nécessaire.

La commande `wapt-get upgradedb` renvoie :

```
WARNING upgrade db aborted: current structure version 20161109 is newer or equal to requested␣
↪structure version 20161109
No database upgrade required, current 20161109, required 20161109
```

35.2.7 wapt-get setup-tasks - wapt-get enable-tasks - wapt-get disable-tasks

La commande `wapt-get setup-tasks` ajoute des tâches planifiées **update** et **upgrade**

Indication : Cette fonction est utile lorsque que l'on ne souhaite pas utiliser le service **waptservice**, sinon le service **waptservice** s'en chargera.

Pour que cela fonctionne, il faut au préalable avoir configuré les paramètres dans le fichier `wapt-get.ini` :

- *waptupdate_task_maxruntime* ;
- *waptupgrade_task_maxruntime* ;
- *waptupdate_task_period* ;
- *waptupgrade_task_period* ;

Alors :

- The **wapt-get enable-tasks** command will enable scheduled tasks.
- The **wapt-get disable-tasks** command will disable scheduled tasks.

35.2.8 wapt-get add-upgrade-shutdown - wapt-get remove-upgrade-shutdown

- The **wapt-get add-upgrade-shutdown** command adds a **waptexit** local security policy object, enabling the execution of **waptexit** at system shutdown.
- The **wapt-get remove-upgrade-shutdown** command removes the **waptexit** local security policy object, disabling the execution of **waptexit** during system shutdown.

35.2.9 wapt-get inventory

La commande `wapt-get inventory` affiche le contenu de l'inventaire local de la machine au format JSON.

La commande `wapt-get inventory` renvoie :

```
{
  "wapt": {
    "setuphelpers-version": "1.3.8",
    "waptserver": {
      "dnsdomain": "mydomain.lan",
      "proxies": {
        "http": null,
        "https": null
      },
      "server_url": "https://srvwapt.mydomain.lan"
},
...
```

35.2.10 wapt-get update-status

La commande `wapt-get update-status` renvoie l'inventaire local vers le serveur d'inventaire WAPT.

Note : Si un composant matériel a changé sur la machine, cette commande ne remontera pas la nouvelle information au serveur WAPT.

Dans ce cas, il faudra lancer la commande **`inventory`**.

La commande `wapt-get update-status` renvoie :

```
Inventory correctly sent to server https://srvwapt.mydomain.lan.
```

35.2.11 wapt-get setlocalpassword

La commande `wapt-get setlocalpassword` permet de définir un mot de passe local pour installer des paquets WAPT.

La commande `wapt-get setlocalpassword` renvoie :

```
Local password:
Confirm password:
Local auth password set successfully
```

35.2.12 wapt-get reset-uuid

La commande `wapt-get reset-uuid` permet de récupérer le *UUID* de la machine à partir du BIOS et le remonter au serveur WAPT.

La commande `wapt-get wapt-get reset-uuid` renvoie :

```
New UUID: B0F23D44-86CB-CEFE-A8D6-FB8E3343FE7F
```

35.2.13 wapt-get generate-uuid

La commande `wapt-get generate-uuid` permet de générer un *UUID* aléatoire pour la machine et de le remonter au serveur WAPT.

Indication : Certaines machines ont un BIOS avec des *UUID* identiques. Cette anomalie est un défaut de paramétrage du BIOS par le constructeur de la machine car les *UUID* devraient être uniques.

La commande command :*generate-uuid* existe pour résoudre ce problème.

La commande `wapt-get generate-uuid` renvoie :

```
New UUID: 6640f174-de90-4b00-86f7-d7834ceb45bc
```

35.2.14 wapt-get get-server-certificate

La commande `wapt-get get-server-certificate` permet de télécharger le certificat SSL du serveur WAPT, quand on utilise HTTPS pour les échanges avec le serveur.

Le certificat téléchargé est stocké dans `C:\Program Files (x86)\wapt\ssl\server`.

La commande `wapt-get get-server-certificate` renvoie :

```
Server certificate written to C:\Program Files (x86)\wapt\ssl\server\srvwapt.mydomain.lan.crt
```

35.2.15 wapt-get enable-check-certificate

La commande `wapt-get enable-check-certificate` permet de télécharger le certificat SSL du serveur WAPT et active le contrôle des échanges avec le serveur.

La commande `wapt-get enable-check-certificate` renvoie :

```
Server certificate written to C:\Program Files (x86)\wapt\ssl\server\srvwapt.mydomain.lan.crt
wapt config file updated
```

35.2.16 wapt-get session-setup

La commande `wapt-get session-setup` lance l'exécution des personnalisations logicielles en contexte utilisateur.

Indication : Les instructions en **`session-setup`** sont définies dans le fichier `setup.py` des paquets applicatifs.

L'ensemble des instructions est sauvegardé en base locale.

Le script **`session-setup`** est lancé à chaque démarrage mais l'exécution d'un **`session-setup`** ne se fait qu'une fois par utilisateur et par version de paquet.

Note : Le paramètre *ALL* lancera le **`session-setup`** de tous les paquets WAPT installés sur la machine.

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-rsat-x64 ... No session-setup. Done
Configuring tis-dotnetfx4.6 ... No session-setup. Done
Configuring tis-rsat-win7x64 ... No session-setup. Done
Configuring tis-mumble ... No session-setup. Done
Configuring tis-paint.net ... No session-setup. Done
Configuring wsagauvrit.domain.lan ... No session-setup. Done
```

35.3 Utiliser la ligne de commande pour créer des paquets WAPT

35.3.1 wapt-get make-template

La commande `wapt-get make-template <msi or exe file> <package name>` permet de générer un modèle de paquet logiciel à partir d'un installeur exécutable au format MSI ou EXE.

Vous trouverez ici la procédure complète pour *créer des paquets WAPT*.

Indication :

— Si vous avez au préalable installé le paquet `tis-waptdev` sur votre machine de développement, l'éditeur **PyScripter** se lancera automatiquement en ouvrant le projet de création de paquet.

La commande `wapt-get make-template C:UsersUserDownloadstightvnc-2.8.5-gpl-setup-64bit.msi tis-tightvnc` renvoie :

```
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
```

35.3.2 wapt-get make-host-template

La commande `wapt-get make-host-template <host FQDN>` permet de générer un modèle de paquet machine vide.

La commande `wapt-get make-host-template host01.mydomain.lan` renvoie :

```
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
```

35.3.3 wapt-get make-group-template

La commande `wapt-get make-group-template <name of group>` permet de générer un modèle de paquet groupe vide.

La commande `wapt-get make-group-template accounting` renvoie :

```
Template created. You can build the WAPT package by launching
  C:\Program Files (x86)\wapt\wapt-get.exe build-package C:\waptdev\accounting-wapt
You can build and upload the WAPT package by launching
  C:\Program Files (x86)\wapt\wapt-get.exe build-upload C:\waptdev\accounting-wapt
```

35.3.4 wapt-get list-registry

La commande `wapt-get list-registry <keyword>` permet de rechercher un mot clé parmi les logiciels installés sur la machine.

Cette commande affiche un tableau contenant la clé de désinstallation de chaque logiciel correspondant au critère de recherche.

La commande `wapt-get list-registry firefox` renvoie :

```
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"
```

35.3.5 wapt-get sources

La commande `wapt-get sources <package name>` permet de télécharger les sources depuis un dépôt versionné type Git / SVN.

La commande `wapt-get sources tis-firefox` ne renvoie rien ;

35.3.6 wapt-get build-package

La commande `wapt-get build-package <path to the package>` permet de construire le paquet WAPT et le signer avec la clé privée de l'*Administrateur*.

Note : Il convient de s'assurer que le chemin de la clé privée, le préfixe et le chemin de développement par défaut sont renseignés.

The command `wapt-get sources tis-firefox` returns :

```
Building  C:\waptdev\tis-tightvnc-wapt

Package tis-tightvnc (=2.8.5.0-0) content:
 setup.py
 tightvnc-2.8.5-gpl-setup-64bit.msi
 WAPT\control
 WAPT\wapt.psproj
...done. Package filename C:\waptdev\tis-tightvnc_2.8.5.0-0_all.wapt
Signing C:\waptdev\tis-tightvnc_2.8.5.0-0_all.wapt

7-Zip [64] 16.04: Copyright (c) 1999-2016 Igor Pavlov: 2016-10-04

Open archive: C:\waptdev\tis-tightvnc_2.8.5.0-0_all.wapt
--
Path = C:\waptdev\tis-tightvnc_2.8.5.0-0_all.wapt
Type = zip
Physical Size = 1756459

Updating archive: C:\waptdev\tis-tightvnc_2.8.5.0-0_all.wapt
```

(suite sur la page suivante)

(suite de la page précédente)

```
Items to compress: 0

Files read from disk: 0
Archive size: 1755509 bytes (1715 KiB)
Everything is Ok
Package C:\waptdev\tis-tightvnc_2.8.5.0-0_all.wapt signed: signature:
mOQINvKGfmcW4nu05aVc8MJqMtXdPv5I0qo5zCfMkIWvEeYYDDfnZLakPkXiqptiqcNbCdY8vOPs
qFMqwSMYUyKJ8d3DHEk8kdlIldkLsiAejkdsoiZDKlEFVCJgdKI13x4FcPfoZNw5DFPzmCZKbgkU
pWvGbGFwUx/3d9zcliciN82F0FveC6C0mqoh5A==

You can upload to repository with
  C:\Program Files (x86)\wapt\wapt-get.exe upload-package "C:\waptdev\tis-tightvnc_2.8.5.0-0_all.
↪wapt"
```

35.3.7 wapt-get sign-package

La commande `wapt-get sign-package <path to the package>` permet de signer un paquet téléchargé manuellement avec la clé privée de l'*Administrateur* en ligne de commande.

Attention : La commande **`sign-package`** ne renomme pas le paquet avec le préfixe de l'*Organisation*.

La commande `wapt-get sign-package C:\waptdev\smp-7zip_16.4.0.0-1_all.wapt` renvoie :

```
Signing C:\waptdev\smp-7zip_16.4.0.0-1_all.wapt

7-Zip [64] 16.04: Copyright (c) 1999-2016 Igor Pavlov: 2016-10-04

Open archive: C:\waptdev\smp-7zip_16.4.0.0-1_all.wapt
--
Path = C:\waptdev\smp-7zip_16.4.0.0-1_all.wapt
Type = zip
Physical Size = 2857855

Updating archive: C:\waptdev\smp-7zip_16.4.0.0-1_all.wapt

Items to compress: 0

Files read from disk: 0
Archive size: 2856021 bytes (2790 KiB)
Everything is Ok
Package C:\waptdev\smp-7zip_16.4.0.0-1_all.wapt signed: signature:
lAxMJBKlnZLFQG81Rwb80+cB6XHcNjazmVJI7+PLLcPfFkFVC5wojyMPVMKhUrjrSlWomj85L8CY
gZv/FsVspUij45TcikukbF8Rr+jy6saHskg42XINqZWCnP28k4bkIREdzYIkuKDABfr15gt3ecuN
E21ZU/SI8BtXOX/80w9hpbP6ivCzTaYZZk18dhLDzV04xM9QwPSZ2mjQspbVklpm2NL4F6gb5b9D
EwMjus74/MNc6BZeKtMcFcE3Ft18ROAJeF5hLws24jjCv6Gjjus+zlGlepWK0M2p7rIdvmC1BWB/
Y6e1mQpSoisAvhOpATFPqNJca/QTMANKiTD3OA==
```

35.3.8 wapt-get build-upload

La commande `wapt-get build-upload <chemin du paquet>` permet de construire et d'uploader le paquet résultant sur le dépôt WAPT local.

Indication : Avec le paramètres *-i* on incrémente directement la version du paquet WAPT sans avoir à modifier le fichier `control`.

La commande `wapt-get -i build-upload C:\waptdev\tis-tightvnc-wapt` renvoie :

```
Building  C:\waptdev\tis-tightvnc-wapt
Package tis-tightvnc (=2.8.5.0-1) content:
 setup.py
 tightvnc-2.8.5-gpl-setup-64bit.msi
 WAPT\control
 WAPT\wapt.psproj
...done. Package filename C:\waptdev\tis-tightvnc_2.8.5.0-1_all.wapt
Signing C:\waptdev\tis-tightvnc_2.8.5.0-1_all.wapt

7-Zip [64] 16.04: Copyright (c) 1999-2016 Igor Pavlov: 2016-10-04

Open archive: C:\waptdev\tis-tightvnc_2.8.5.0-1_all.wapt
--
Path = C:\waptdev\tis-tightvnc_2.8.5.0-1_all.wapt
Type = zip
Physical Size = 1756458

Updating archive: C:\waptdev\tis-tightvnc_2.8.5.0-1_all.wapt

Items to compress: 0

Files read from disk: 0
Archive size: 1755509 bytes (1715 KiB)
Everything is Ok
Package C:\waptdev\tis-tightvnc_2.8.5.0-1_all.wapt signed: signature:
FVn2yx77TwUHaDauSPHxJZiPAyMQe4PqLF5n6wY9YPAwY4ijHe6NgDFrexXf8ZYbHAiNa5b8V/Qj
wTVHiqpbXnZotiVIGrJDhgbaLwZ9CK6pfWiflC4126nx6PMF3T1i6w0R0NOE2wJpOSRYESk7lDUz
9CPfzJCLcOXwh0F5eZc96wbkDkSbpn1f+x5tOlvyy/FW2m8RbZQhJcO21j9gGX7It0QNecaOxXgz
qkZZKBDNASOBYAF22M1+zHb59DWQ63Q8yMj5t5szEUTkGtQNG6vZz3gb9Yraq361BIGaBDYUM31j
ZgpaHvP0vdK3c1x1mhyhC7q6eZ/UCW5tETTCiA==

Uploading files...
WAPT Server user :admin
WAPT Server password:
Status: OK, tis-tightvnc_2.8.5.0-1_all.wapt uploaded, 1 packages analysed
```

35.3.9 wapt-get duplicate

La commande `wapt-get duplicate <package source> <package new_duplicate>` permet de dupliquer localement un paquet WAPT du dépôt.

La commande `wapt-get duplicate tis-firefox tis-firefox-custom` renvoie :

```
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
```

35.3.10 wapt-get edit

La commande `wapt-get edit <package name>` permet d'éditer un paquet.

La commande `wapt-get edit tis-firefox` renvoie :

```
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\tis-firefox-wapt
```

35.3.11 wapt-get edit-host

La commande `wapt-get edit-host <host FQDN>` permet d'éditer un paquet *host* depuis le dépôt WAPT.

35.3.12 wapt-get upload-package

La commande `wapt-get upload-package <path to the package>` permet de charger un paquet sur le dépôt WAPT.

La commande `wapt-get upload-package C:\waptdevis-tightvnc_2.8.5.0-1_all.wapt` renvoie :

```
WAPT Server user :admin
WAPT Server password:
tis-tightvnc_2.8.5.0-1_all.wapt uploaded, 1 packages analyzed
result: OK
```

35.3.13 wapt-get update-packages

La commande `wapt-get update-packages <path to folder>` permet de scanner un dossier local et de créer un fichier d'index `Packages`.

La commande `wapt-get update-packages D:\Data\WAPT` renvoie :

```
Packages filename: D:\waptdev\Packages
Processed packages:
  D:\Data\WAPT\groupe_base.wapt
  D:\Data\WAPT\tis-firefox_50.1.5.0-0_all.wapt
  D:\Data\WAPT\tis-tightvnc_2.8.5.0-1_all.wapt
  D:\Data\WAPT\tis-7zip_16.4.0.0-1_all.wapt
```

(suite sur la page suivante)

(suite de la page précédente)

```
  D:\Data\WAPT\tis-mumble_3.14-3_all.wapt
  D:\Data\WAPT\tis-noforcereboot_1.0-1_all.wapt
Skipped packages:
```

Re-Signer les paquets sur le serveur en ligne de commande

Danger : Avant d'utiliser cette méthode, assurez-vous que votre serveur WAPT est sécurisé et n'est pas sous contrôle d'une entité tierce non autorisée.

— Copiez votre *.crt* et *.pem* vers `/tmp/` sur le serveur WAPT en utilisant **Winscp** ou un logiciel equivalent.
— Il est alors possible de re-signer tous les paquets en une 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/
```

Indication : Use this method if re-signing from the WAPT console method does not complete successfully.

Attention : Supprimez votre *.crt* et *.pem* de `/tmp/` sur le serveur WAPT.

Pour créer vos paquets et les personnaliser, suivez cette documentation et vous maîtriserez très vite les subtilités du fonctionnement de WAPT.

CHAPITRE 36

Structure détaillée d'un paquet WAPT

Un paquet WAPT est un fichier *.zip* contenant plusieurs élements :

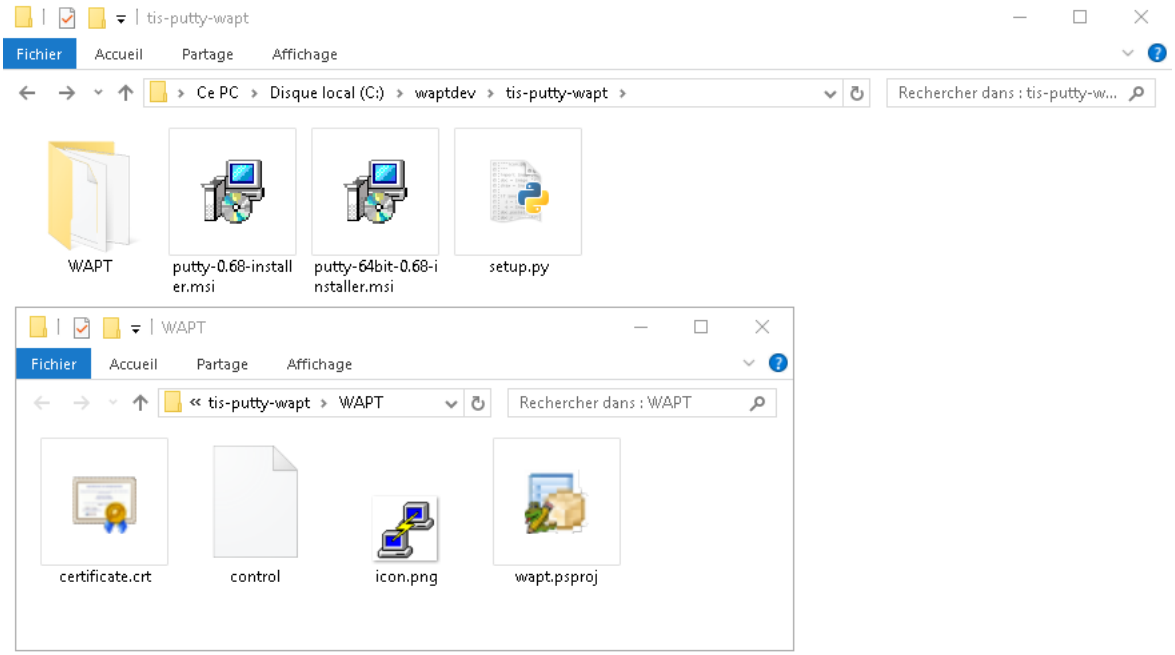


FIG. 1 – Structure d'un paquet WAPT

— 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.psproj` dans le dossier `WAPT`, ce fichier est utilisé pour stocket 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 ;

36.1 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	Example 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 Statisques et la 3D
description_fr	La description avec la langue localisée du paquet	Calculatrice graphique
depends	Les paquets qui doivent être **installés** avant l'installation du paquet	tis-java
conflicts	Les paquets qui doivent être **désinstallés** avant l'installation du paquet	tis-graph
maturity	Niveau de Maturité (*BETA*, *DEV*, *PROD*)	PROD
locale	La langue de l'environnement pour le paquet	fr,en,es
target_os	Les Système d'exploitation compatibles avec le paquet	windows,mac,linux
min_os_version	La version minimale de Windows pour que la paquet soit vu par l'agent WAPT	6.0
max_os_version	La version maximale de Windows pour que la paquet soit vu par l'agent WAPT	10.0
min_wapt_version	La version minimum de WAPT pour que la paquet fonctionne correctement	1.3.8
sources	Chemin vers l'emplacement SVN du paquet (**source** command)	https://srv-svn.mydomain.lan/sources/tis-geogebra-wapt/trunk/
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	Si officiel du logiciel contenu dans le paquet WAPT	https://www.tranquil.it/
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-Liz0qCHN5fChdylnvXUZ8xNJj4rEu5FA
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

Attention : Si le fichier `control` contient des caractères spéciaux, le fichier `control` doit être sauvegardé en format **UTF-8 (No BOM)**.

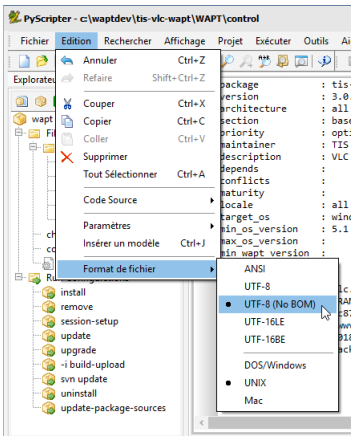


FIG. 2 – PyScripter - UTF-8 (No BOM)

36.1.1 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 équippés 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 descritpion 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 séprant par des virgules (,).

```
depends: tis-java,tis-firefox-esr,tis-thunderbird
```

conflicts

Fonctionne de manière opposé à *depends*.

Les *conflits* 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 séprant par des virgules (,).

```
conflicts: tis-firefox
```

maturity

Définit la maturité d'un paquet.

Par défaut, les agents WAPT verront des paquet 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 language 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 utilsé 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 utilsé 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 utilsé 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 ldisque 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.

36.2 Le fichier *setup.py*

36.2.1 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 dévleopper plus facilement des paquets entièrement fonctionnels.

36.2.2 uninstallkey list

Nous trouvons alors :

```
uninstallkey = ['tisnaps2','Mozilla Firefox 45.6.0 ESR (x86 fr)']
```

Ici, nous associons une list 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 assoiciée 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 = []
```

36.2.3 Fonction install()

Viens alors la déclaration des fonctions dans `setup.py`.

C'est la recette du paquet WAPT, l'ensemble d'instruction qui sera exécuté.

```
def install():
    run('"install.exe" /S')
```

36.3 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.

36.4 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.

36.5 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.

36.6 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` ;

36.7 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 ;

CHAPITRE 37

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 – The principal syntax differences

Syntaxe	Python 2	Python 3
print	`print'Hello'`	`print('Hello')`
unicode string	`ur`	`r`
opérateurs	`<> <=> !=`	`!=`
Acces à la base de registre Windows	`_winreg`	`winreg`

Indication : For more details, visit :

— https://python-future.org/compatible_idioms.html.

— https://blog.couchbase.com/tips-and-tricks-for-upgrading-from-python-2-to-python-3/.

CHAPITRE 38

Créer son environment de développement de paquets WAPT

38.1 Pré-requis

Attention :

- It is **required** to be a *Local Administrator* of the machine to use WAPT's integrated environment for developing WAPT packages.
- We advise you to create/ edit packages in a fully controlled environment that is safe and *disposable*.
- The usage of a disposable virtual machine (like Virtualbox) is recommended.
- Import the *tis-pyscripter* package in your local repository and install it on your development computer.

38.2 Préconisations concernant l'environment 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'environments afin qu'il devienne le plus abouti possible en régime de test, avant d'être basculé en production.

38.3 Démarche de test

38.3.1 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 : When possible, RC and Beta version of Operating Systems should be tested.

38.3.2 L'état des mises à jour Windows

— **Microsoft Windows host without any Windows update installed** : the objective is to detect Windows updates that are required for the software to work properly and to adapt the WAPT package accordingly ;
— **Microsoft Windows host with all the latest Windows updates** : the objective is to detect Windows updates that break the package and to adapt the WAPT package accordingly ;

38.3.3 Etat des installations des logiciels

— **Hosts with many installed packages** : the objective is to detect a possible dependency with an existing application ;
— **Hosts with many installed packages** : the objective is to detect a possible conflict with an existing application ;
— **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 ;

CHAPITRE 39

Principles of creating package template from the WAPT console

Indication : Pour créer des paquets à partir de la console, il faut d'abord avoir installé l'environnement de développement WAPT *tis-pyscripter*.

39.1 Créer un paquet WAPT depuis la console

Dans cet exemple, l'installeur de 7zip est utilisé au format MSI.

- Download 7-zip MSI x64.
- Create a WAPT package Template from the installer.
  Dans la console WAPT, cliquer sur *Outils* → *Créer un modèle de paquet depuis un installeur* :
  Sélectionner l'installeur MSI téléchargé et renseigner les différentes informations demandées. Veillez bien à ce que le nom du paquet ne contienne pas de numéro de version.
- Two solutions are available :
  - Click on *Make and edit ….* (recommended) to verify the WAPT package and customize it to your Organization's specific needs.
  - Click on *Build and upload* to directly build and upload the package into your private repository.

  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.
  However, the first method that consists of first testing locally the package before uploading is the recommended method.

Note : *An old command line method is also available*.

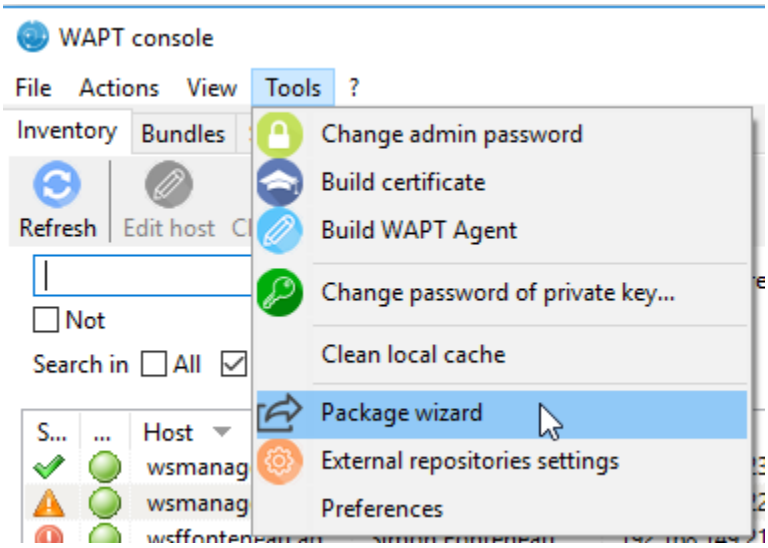


FIG. 1 – PyScripter - WAPT console window for creating a package template

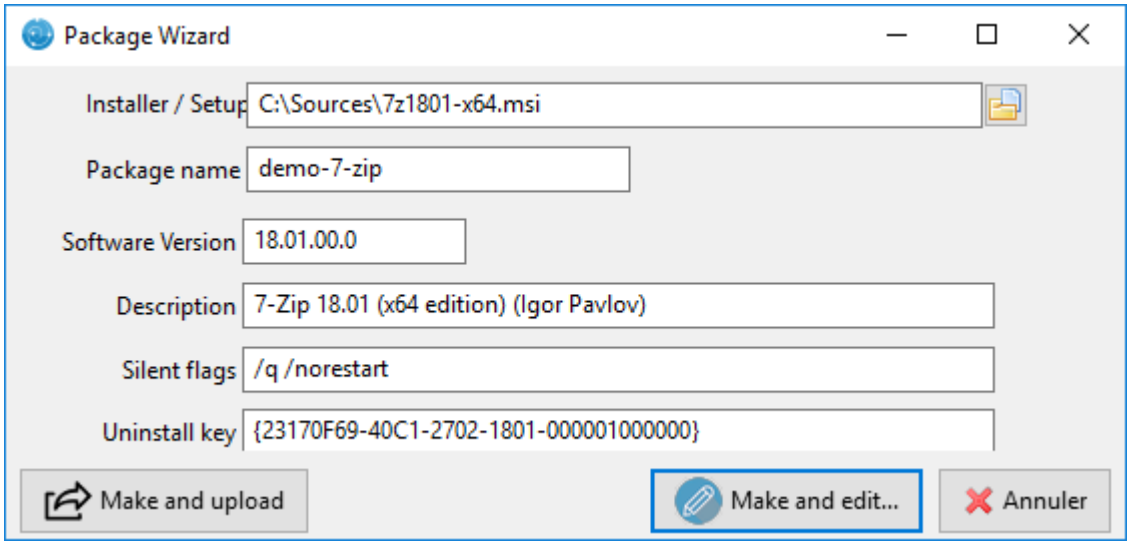


FIG. 2 – PyScripter - Renseignements nécesssaires pour la création du paquet

39.2 Customizing the package before uploading it to your repository

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*.

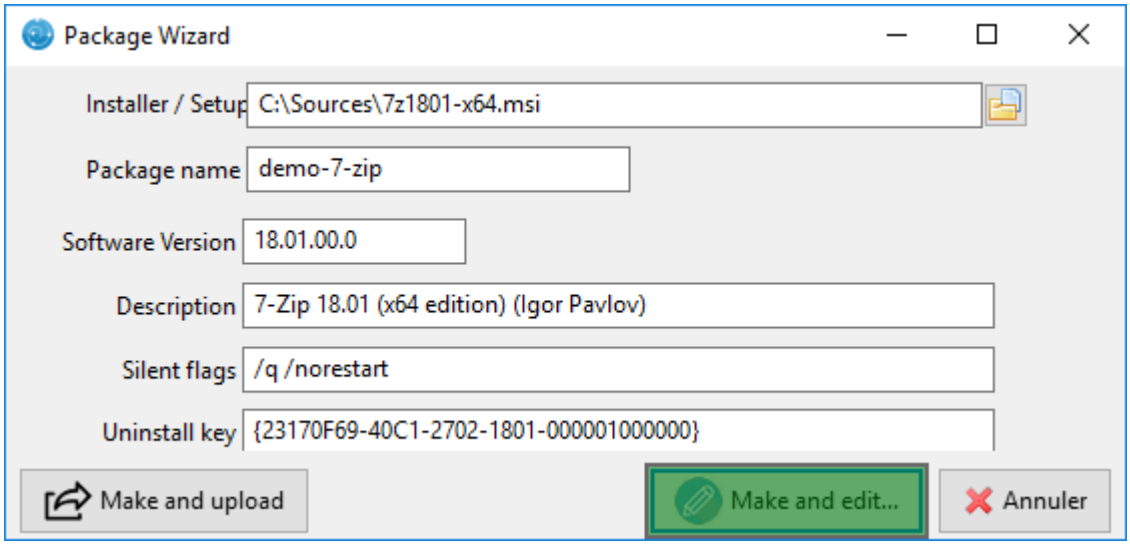


FIG. 3 – PyScripter - Renseignements nécésssaires pour la création du paquet

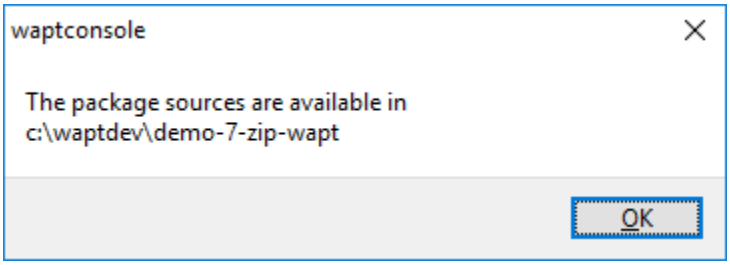


FIG. 4 – PyScripter - Validation de la création du modèle et ouverture de PyScripter

L'IDE **PyScripter** se lance et permet d'éditer les fichiers du paquet.

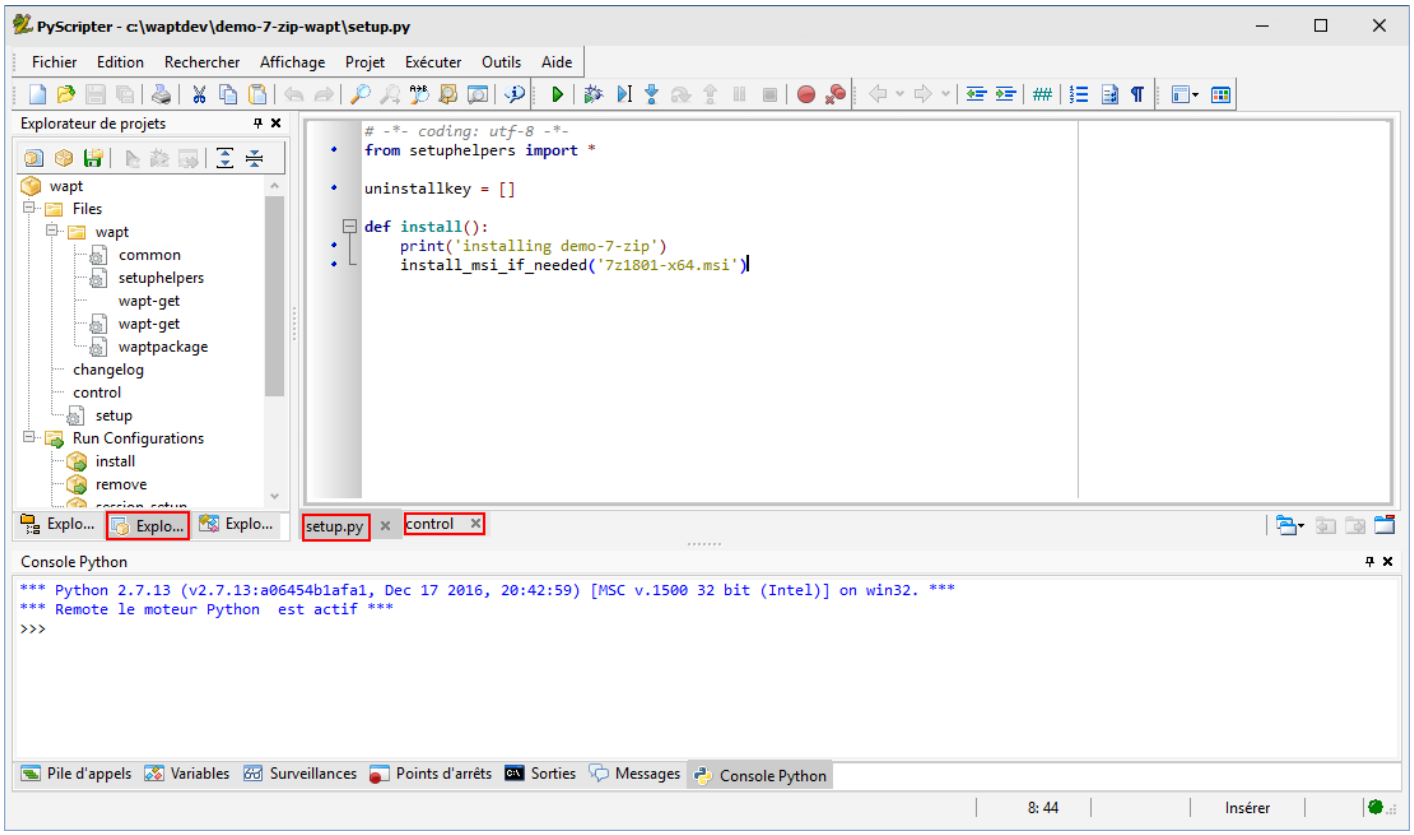


Fig. 5 – PyScripter - Customizing a package with PyScripter

CHAPITRE 40

Presentation of PyScripter

40.1 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`.

40.2 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.

40.3 Zone d'édition

La Zone d'édition de **PyScripter** permet d'éditer le fichier `setup.py` ainsi que le fichier `control`.

40.4 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`.

To learn more about the composition of a WAPT package, visit the documentation on the structure of a WAPT package.

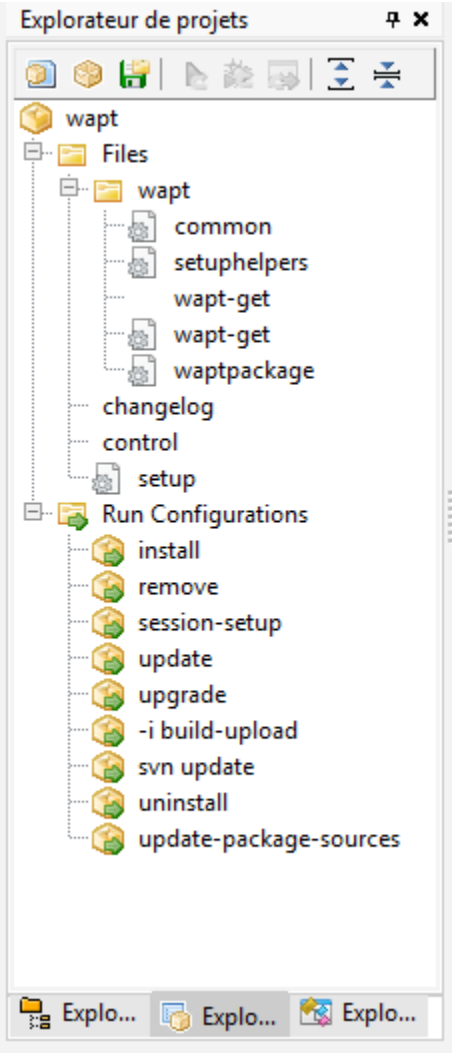


FIG. 1 – PyScripter - Explorateur de projets PyScripter

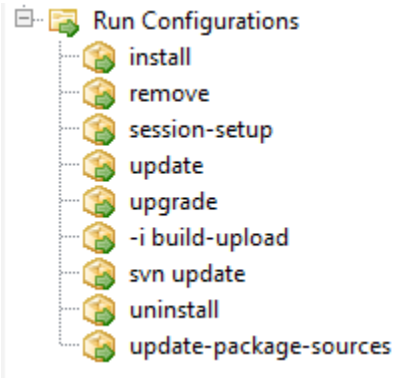


FIG. 2 – PyScripter - Commandes Run dans l'explorateur de projets PyScripter

```
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

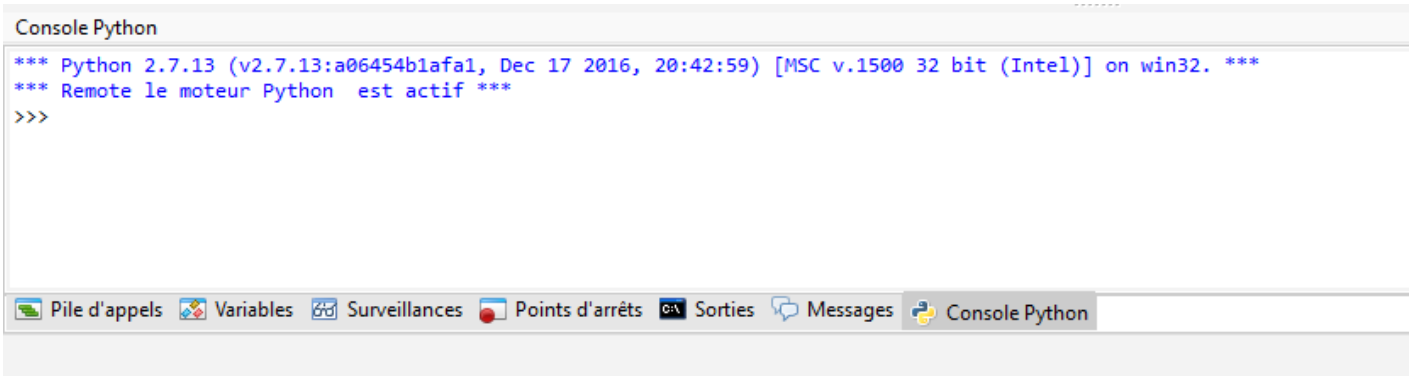


FIG. 4 – PyScripter - console python de PyScripter

40.5 Tester localement l'installation du paquet WAPT

Vous pouvez ensuite tester le lancement d'une installation sur votre station de développement.

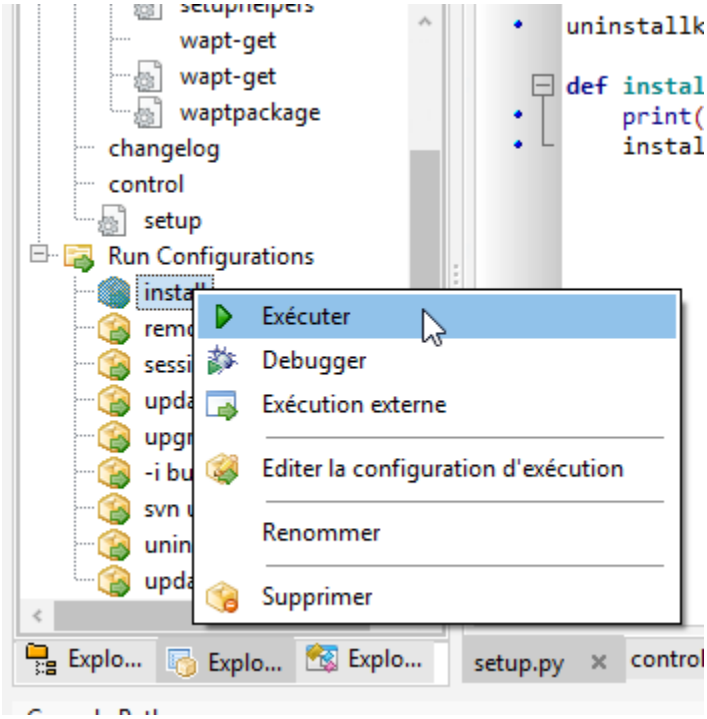


Fig. 5 – Running an install command from the PyScripter console

La Console PyScripter vous permet de vérifier si l'installation s'est bien déroulée.

40.6 Tester localement la désinstallation du paquet WAPT

Vous pouvez ensuite tester le lancement d'une désinstallation sur votre station de développement.

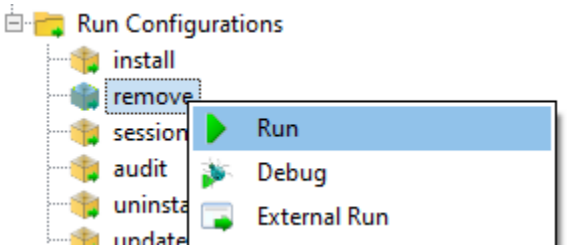


Fig. 6 – Running a remove command from the PyScripter console

La Console PyScripter vous permet de vérifier si l'installation s'est bien déroulée.

CHAPITRE 41

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*.

Edit the `control` file (`architecture`, `impacted_process`, `target_os`, `description`, `maintainer` ...). For more information, visit the documentation on the control file structure.

Your **PyScripter** opens, go to your `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')
```

— The function will test whether a version of the software is already installed on the host using the *uninstall key*.
— If the *uninstall key* is already present, the new version of the software will be installed only if the installed version is older.
— After the installation, the function will finally test that the *uninstall key* is present and its version, to ascertain that all went well.

Tableau 1 – Liste des arguments disponibles avec *install_exe_if_need*

Paramètres	Valeur par défaut	Description
`msi`		nom du fichier *.msi* 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 *installed_softwares*
`remove_old_version`	False	supprimme automatiquement une ancienne version d'un logiciel dont la *uninstallkey* est identique
`force`	False	force l'installation du logiciel même si une *uninstall key* 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)
```

41.1 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')
```

41.2 Vidéo de démonstration

CHAPITRE 42

Packager des .exe (exemple)

- Download the *.exe* installer from a reliable source.
  Téléchargez l'installateur au format exe Firefox ESR x64 sur https://download.mozilla.org/?product=firefox-esr-latest-ssl&os=win64
- Look up documentation relating to silent flags :
  - On the Official Mozilla website.
  - Other methods for finding information on silent flags :
    - WPKG packages repository.
    - Chocolatey packages repository.
    - Search on the Internet with the search terms : *Firefox silent install*.
- Then generate your package template, please refer to the *documentation for creating packages from the WAPT console*. **PyScripter** loads up and opens the *.exe* package project.
- Edit the `control` file (`architecture`, `impacted_process`, `target_os`, `description`, `maintainer` ...). For more information, visit the documentation on the control file structure.
- Check the `control` file content. Mozilla Firefox-ESR does not comply to industry standards and returns an erroneous version number (it appears to be the installer packaging software version number).
  - Original `control` file.

```
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  :
```

  - Modified `control` file.

```
package           : tis-firefox-esr
version           : 52.6.0-1
```

(suite sur la page suivante)

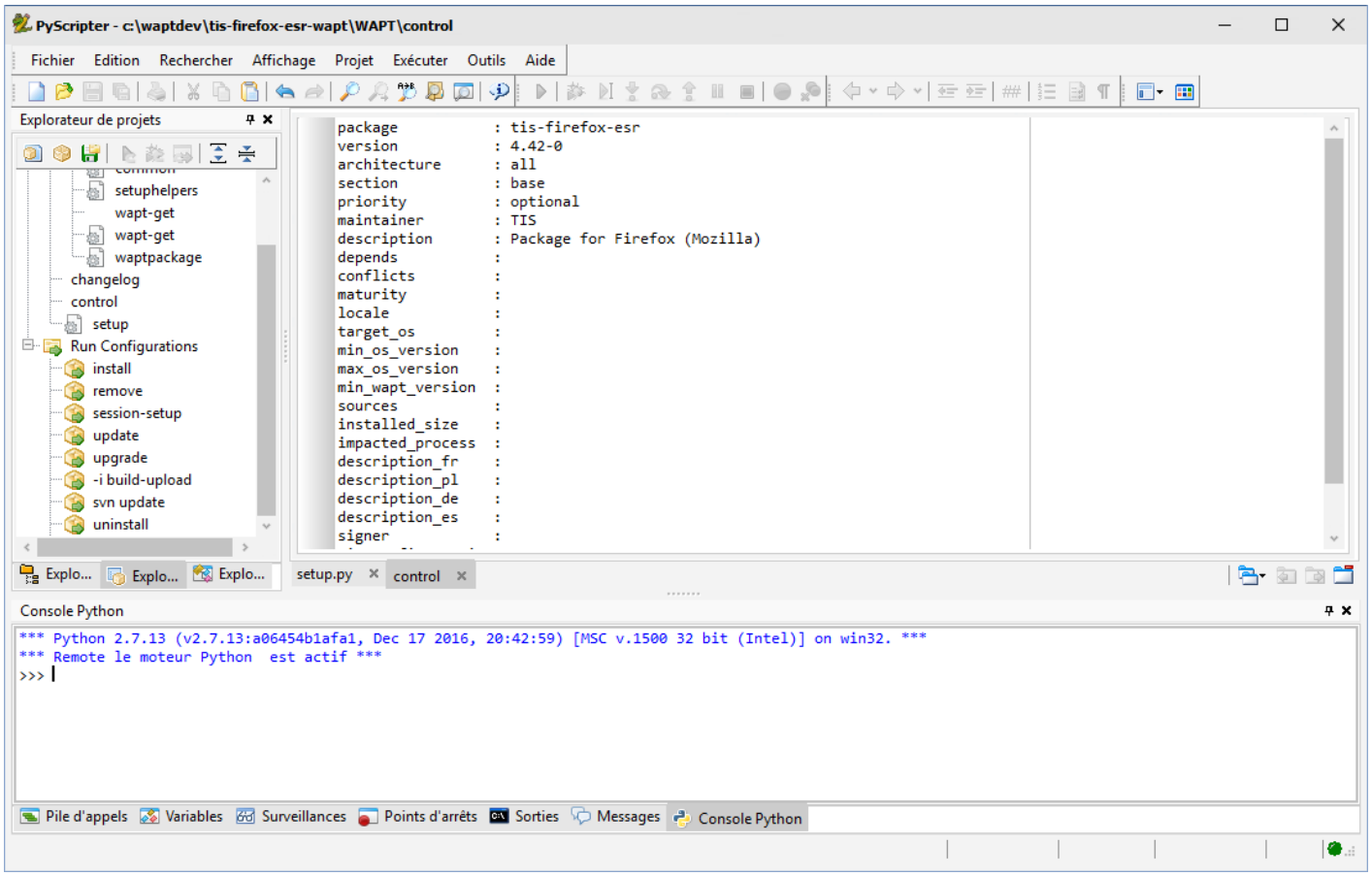


FIG. 1 – PyScripter affichant le fichier *control*

(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 : A sub-version *-1* has been appended to the software version number ; it is the packaging version of the WAPT package. 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 :

- The function requires to pass the silent flag as an argument.
- The function requires to pass the *uninstall key* as an argument.

```
# -*- 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
`exe`		Name of the `.exe` file to execute.
`silentflags`		Silent parameters to pass as arguments to the installer.
`key`	None	Software *uninstall key*.
`min_version`	None	Minimal version above which the software will update.
`killbefore`	[ ]	List of programs to kill before installing the package.
`accept_returncodes`	[0,3010]	Accepted codes other than 0 and 3010 returned by the function.
`timeout`	300	Maximum installation wait time (in seconds).
`get_version`	None	Value passed as parameter to control the version number instead of the value returned by the **`installed_softwares`** function.
`remove_old_version`	False	Automatically removes an older version of a software whose uninstallkey is identical
`force`	False	Forces the installation of the software even though the same uninstallkey has been found

Le paquet aura alors ce comportement :

- Firefox will install only if Firefox is not yet installed or if the installed version of Firefox is less than 45.5.0, unless the `--force` option is passed as argument when installing the package.
- On installing, the running **`firefox.exe`** processes will be killed (with the value indicated in `impacted_process` of the `control` file).
- The function will add by itself the *uninstall key*, so leave the *uninstall key* argument empty.

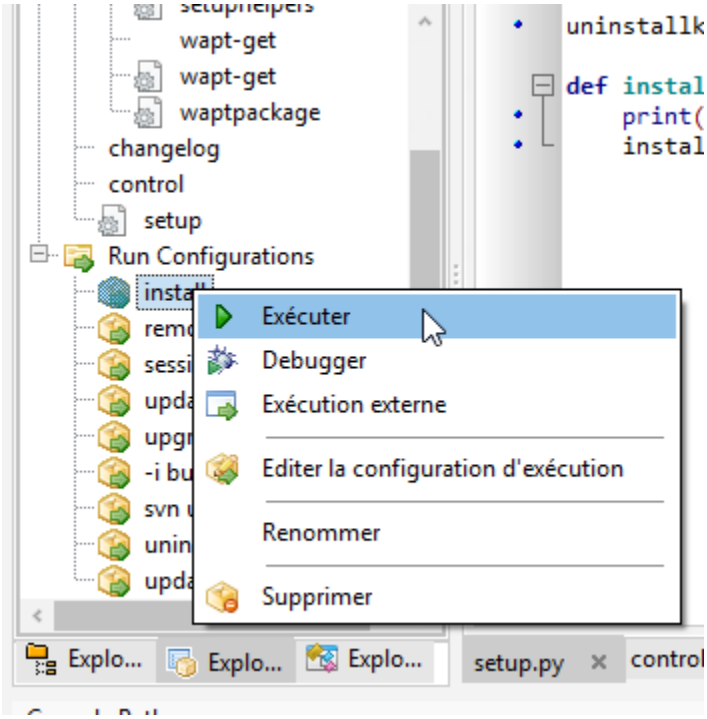
— When finishing the installation of the package, the function will check that the *uninstall key* is present on the machine and that the version of Firefox is greater than 45.5.0 ; if this not the case, the package will be flagged as **ERROR**.

42.1 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.

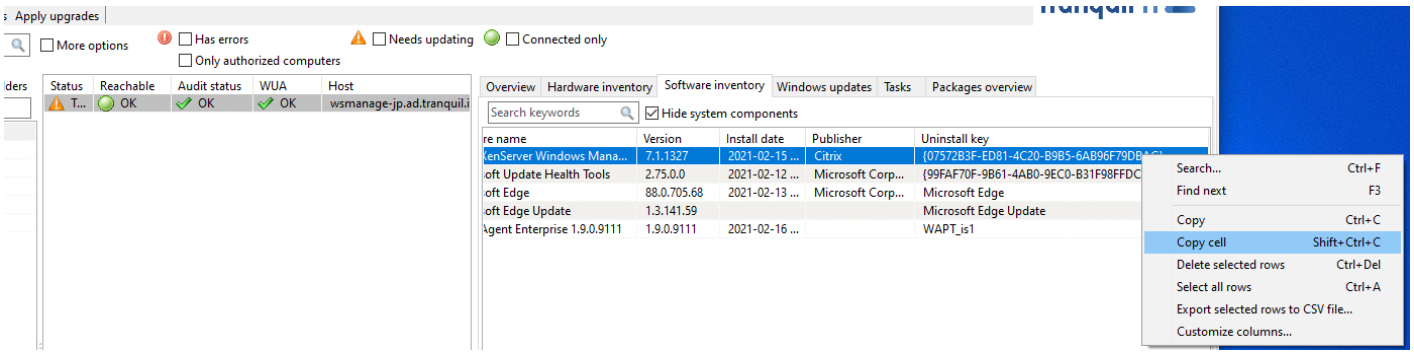


Fig. 2 – Récupérer une clé de désinstallation depuis la console

Vous devez également vérifier la valeur de la version avec la valeur indiquée dans `min_version` dans votre `setup.py`.

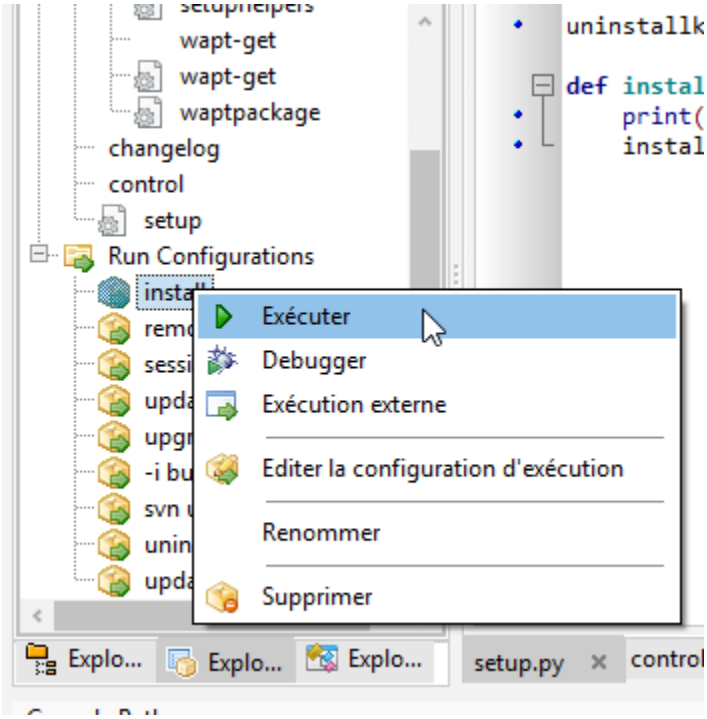
Modifier votre fichier `setup.py` avec les nouveaux paramètres :

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

uninstallkey = []

def install():
    print('installing tis-firefox-esr')
    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 **pyscripter**.



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
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 :

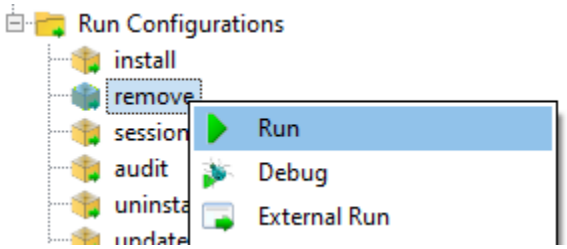


FIG. 3 – Testing the uninstallation

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.

42.2 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 désinstalleur 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 ...

42.3 Vidéo de démonstration

CHAPITRE 43

Packaging empty packages

À faire : gsegat

CHAPITRE 44

Construire le paquet et l'envoyer au serveur WAPT

— Once the package is ready, build it and send it to the WAPT server.

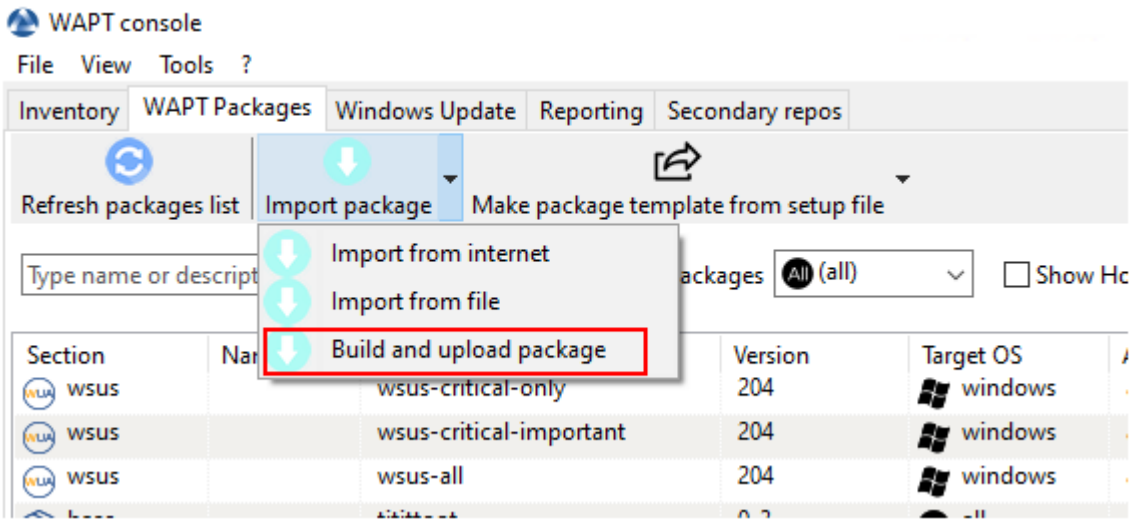


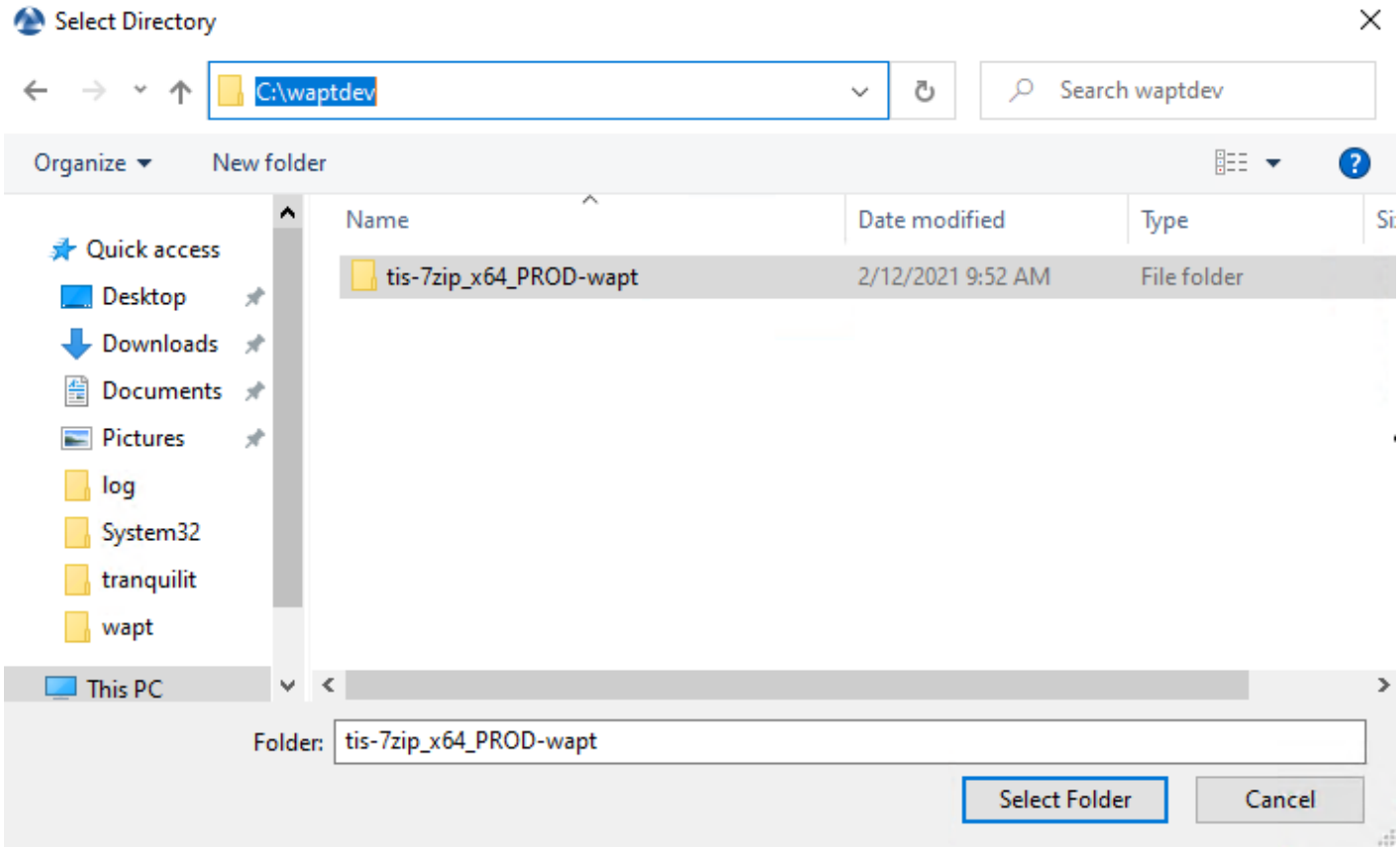
FIG. 1 – Option « Construire et charger le paquet » dans la console WAPT

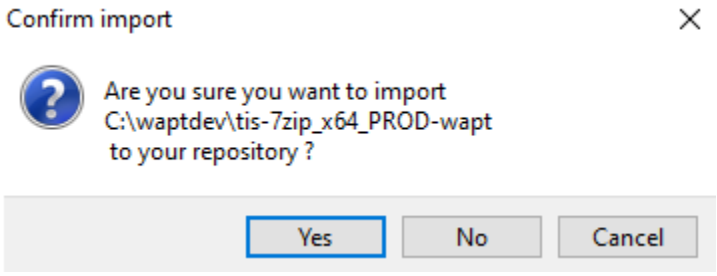
— Select the package in the `c:\waptdev` folder.

— Confirm the selected package.

You have just uploaded your first wapt package.

Note : *Une ancienne méthode en ligne de commande est disponible ici.*


Select Directory
C:\waptdev
Search waptdev
Organize
New folder
Quick access
Desktop
Downloads
Documents
Pictures
log
System32
tranquilit
wapt
This PC
Name
Date modified
Type
tis-7zip_x64_PROD-wapt
2/12/2021 9:52 AM
File folder
Folder:
tis-7zip_x64_PROD-wapt
Select Folder
Cancel


Confirm import
Are you sure you want to import
C:\waptdev\tis-7zip_x64_PROD-wapt
to your repository ?
Yes
No
Cancel

Avertissement : Once your package has uploaded, refresh the package list using the *Refresh packages list* button or by pressing `F5` on your keyboard.

44.1 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>`_.

CHAPITRE 45

Exemples simples de fonctions du setuphelper couramment utilisées

Nous présentons ici quelques fonctions implémentées dans *Setuphelpers* et fréquemment utilisées pour développer des paquets WAPT.

45.1 Tests et manipulation de dossiers et fichiers

45.1.1 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

45.1.2 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

45.1.3 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

45.1.4 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

---

45.1.5 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

---

45.1.6 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

45.1.7 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

---

CHAPITRE 46

Manipulation de clés de registre

46.1 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

46.2 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

46.3 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

CHAPITRE 47

Créer et supprimer des raccourcis

47.1 Un raccourci simple

À faire : gsegat

47.2 Un raccourci dans le menu démarrer

À faire : gsegat

47.3 Créer et supprimer des raccourcis

À faire : gsegat

47.3.1 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

---

47.3.2 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

Firefox place un raccourci sur le bureau de tous les utilisateurs, nous allons le supprimer.

Nous utiliserons la fonction `remove_desktop_shortcut` :

— Modify your `setup.py` and use the function like this.

```
# -*- 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 you restart the installation from **pyscripter**, you will notice that the « all users » desktop shortcut has disappeared.

47.4 Créer un raccourci personnalisé sur le bureau de l'utilisateur en cours

47.4.1 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

47.4.2 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

CHAPITRE 48

Environnement Windows / Logiciel / Services

48.1 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/setuphelpers.html?highlight=windows_version#setuphelpers.windows_version

Visitez également le site sur les numéros de version de Windows.

48.2 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

48.3 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())
```

48.4 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

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

48.5 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

Gère le contexte pour désactiver temporairement le redirecteur wow3264

48.6 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

48.7 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

48.8 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

48.9 Les actions sur les logiciels installés

48.9.1 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

48.9.2 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

48.9.3 Désinstaller des logiciels

```
for soft in installed_softwares('winscp3'):
    if Version(soft['version']) < Version('5.0.2'):
        run(WAPT.uninstall_cmd(soft['key']))
```

— For each item of the list return by *installed_softwares* containing keyword *winscp*.
— If the version is lower than 5.0.2.
— Then uninstall using the *uninstall_cmd* and specifying the corresponding *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

48.9.4 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

CHAPITRE 49

Utiliser les champs du fichier control

Il est possible d'utiliser les informatoins du fichier control dans le `setup.py`

49.1 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.

49.2 Récupérer le nom du logiciel

À faire : Get software name (gsegat)

CHAPITRE 50

Managing a WAPT package with another WAPT package

50.1 Installer un paquet

La commande **install** ...

```
WAPT.install('tis-scratch')
```

... installe *tis-scratch* sur la machine.

50.2 Supprimer un paquet

La commande **remove** ...

```
WAPT.remove('tis-scratch')
```

... désinstalle *tis-scratch* de la machine.

50.3 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*.

CHAPITRE 51

Améliorer mon paquet

51.1 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.

51.2 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 :

— To search unwanted version of **Firefox** we will use the `installed_softwares` function. This function returns a dictionary list containing the software properties :

```
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
      }
]
```

— Check the name of each software.

```
for uninstall in installed_softwares('Mozilla Firefox'):
    print(uninstall['name'])
```

— Show the name of each software found.

```
for uninstall in installed_softwares('Mozilla Firefox'):
    if not 'ESR' in uninstall['name']
        print(uninstall['name'])
```

— Show the name of each software found which does not include the string *ESR* in its name and its uninstallkey.

```
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 accepts an uninstall key as an argument and will send the command to run to start the silent uninstall.

```
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)
```

— Start the uninstallation.

```
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)
```

We can also uninstall the Mozilla maintenance service :

```
for uninstall in installed_softwares('MozillaMaintenanceService'):
    run(uninstall_cmd(uninstall['key']))
```

— Finally, modify your `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**.

51.3 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()
```

51.4 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 :

- Creating user desktop shortcut with specific arguments.
- Making changes to user Windows registry keys.
- Making changes to files, to browser settings of the user.
- Configuring shortcuts to the Organization's set of templates for Documents, Spreadsheets or Presentations in Office Suites to encourage or insure that editorial and graphical guidelines are followed.
- Setting up the user's email or instant messaging from the Organization's main user data repository (LDAP directory, database, etc).
- Customizing an office suite or business software based on the Organization's main user data repository (LDAP directory, database, etc).

La fonction **session_setup** bénéficie de toute la puissance et de l'étendue des librairies python pour atteindre un niveau d'automatisation élevé.

51.4.1 Les principes du *session_setup*

The WAPT **session_setup** function is executed for each user using :

```
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
```

51.4.2 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`).

51.4.3 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.
↪0 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")
```

— Now start the `session-setup` directly from **`pyscripter`**.
— Finally, check that the icon is present on the desktop.

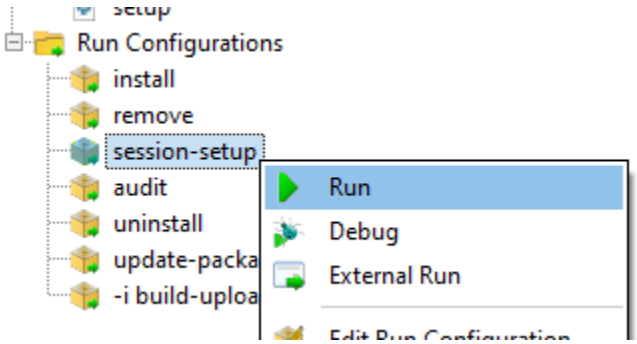


FIG. 1 – PyScripter - running session-setup

51.5 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 :
- Regularly check the list of local administrators on the desktops.
- Ascertain over time the correct configuration of a critical software.
- Regularly check the presence of the correct version of a piece of software.
- Ascertain the security settings of a workstation.

La fonction **audit** bénéficie de toute la puissance et de l'étendue des librairies python pour atteindre une précision d'audit élevée.

51.5.1 Principe de fonctionnement

The **audit** tasks are launched once after every **upgrade**, then regularly as defined by the `audit_schedule` attribute.

Pour executer 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()`** :

In this example, we are improving the Firefox package previously studied in this documentation.
- Add the `audit` function in the `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"
```

— Start the audit from **pyscripter**.

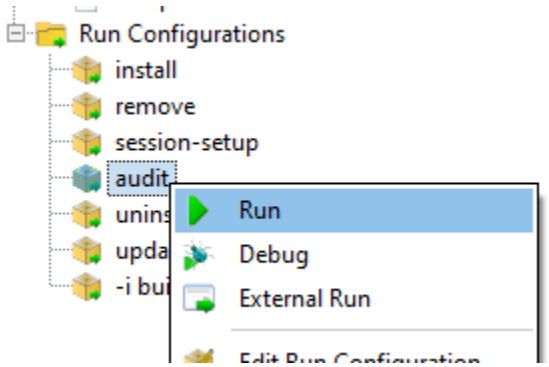


FIG. 2 – PyScripter - Running an audit

— Test with the file then delete the `C:\Program Files\Mozilla Firefox\distribution\policies.json` file and test again with **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.

51.5.2 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 défault 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 :

— An integer (in minutes).
— An integer followed by a letter (m = minutes, h = hours , d = days , w = weeks).

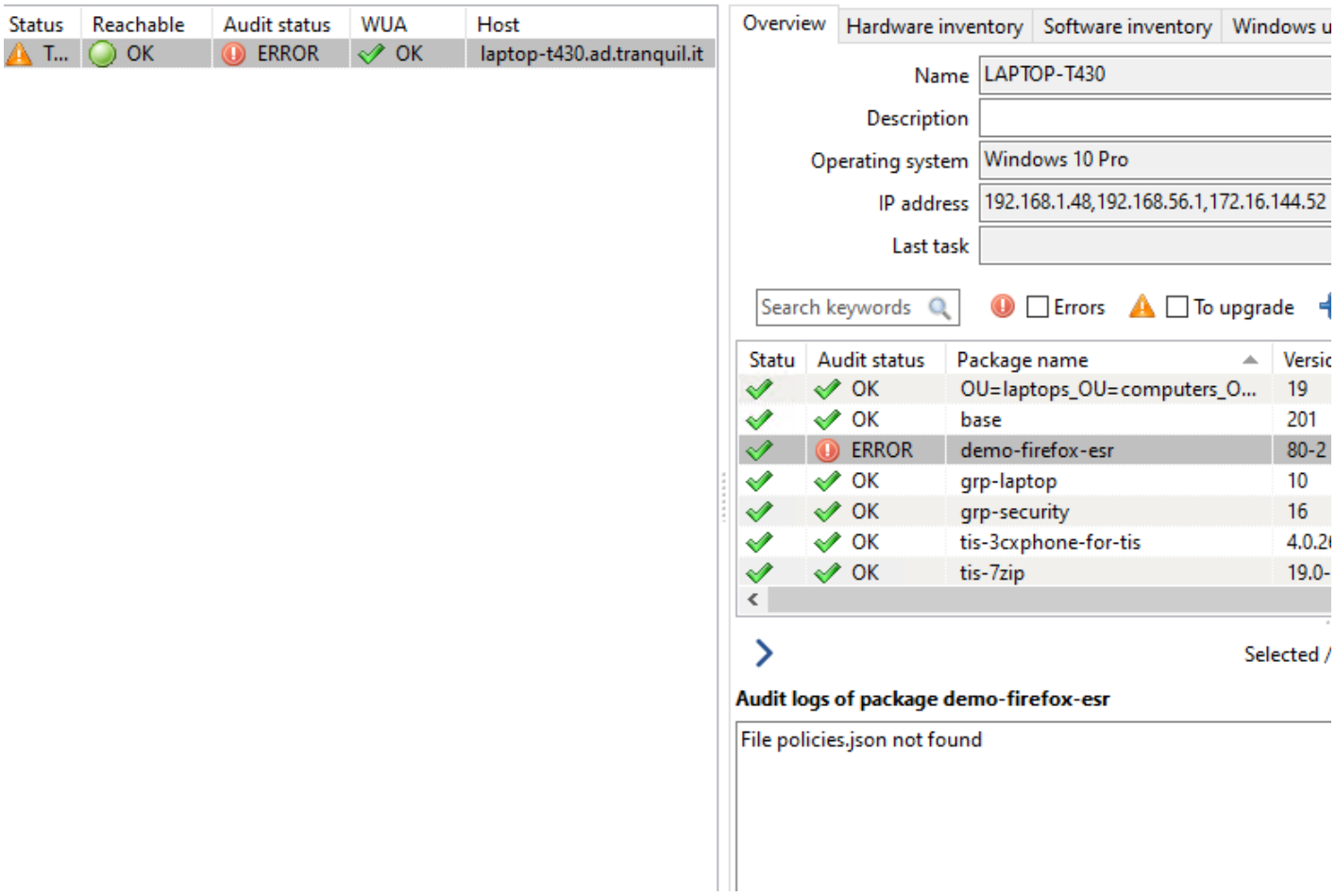


FIG. 3 – Vérifier l'état d'un audit dans la console WAPT

51.5.3 Comportement par défaut de la fonction d'audit

Par défault, 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.

51.6 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.

51.6.1 Principe de fonctionnement

La fonction *update_package* paquet ira :
- Fetch online the latest version of the software.
- Download the latest version of the software binaries.
- Remove old versions of the software binaries.
- Update the version number of the software in the `control` file.

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**.

51.6.2 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)

(suite de la page précédente)

```
    # 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** :

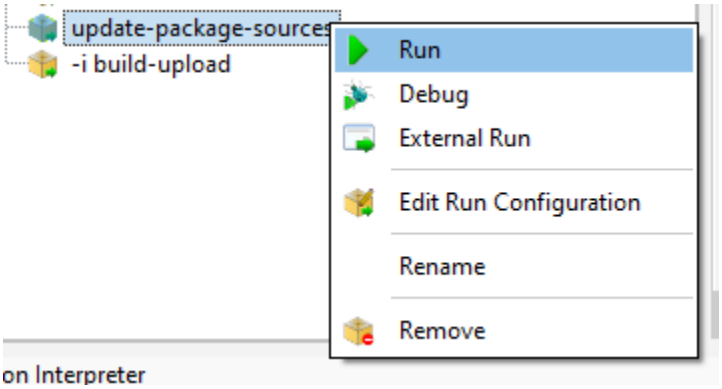


FIG. 4 – PyScripter - Running an update-package-source

Vous trouverez de nombreux exemples d'*update_package* qui vous inspireront dans les paquets du store de Tranquil IT.

51.7 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 :

- Create the folder for the software in `C:\Program Files (x86)`.
- Copy the software in that folder.
- Create the shortcut to the application.
- Manage the uninstallation process for the application.
- Close the application if it's running.

51.7.1 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()
```

51.8 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.

- Download KB4522355 MSU package from Microsoft Catalog website.
- Create a WAPT package template from the downloaded *.msu* file. In the WAPT console, click on *Tools* → *Package Wizard*.
- Select the downloaded *.msu* package and fill in the required fields.
- Click on *Make and edit* (recommended) to launch package customization.
- WAPT package IDE is launched using the source code from the pre-defined *.msu* template.
- As usual with WAPT packages, test, then build, then sign, then upload and finally affect the desired WAPT packages to your selected hosts and it is done ! !
- If the KB becomes bundled with the following *Patch Tuesday*, you can select the hosts onto which the package has been applied and forget the KB package on the hosts.

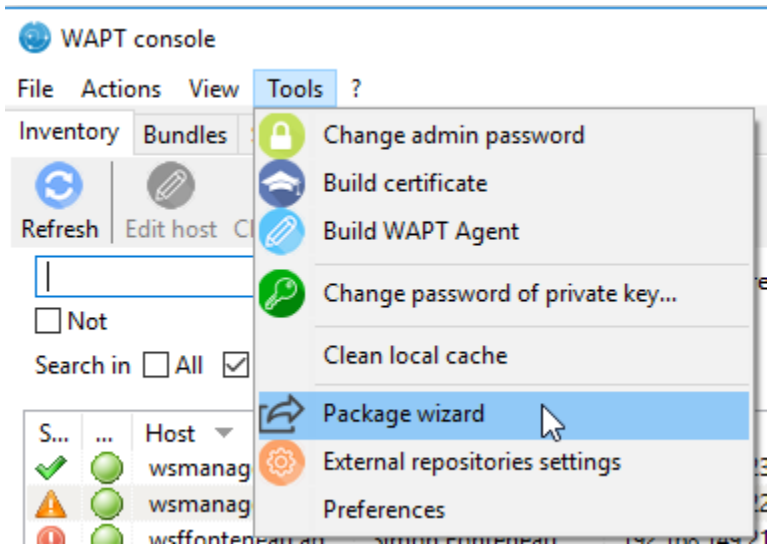


FIG. 5 – PyScripter - WAPT console window for creating a package template

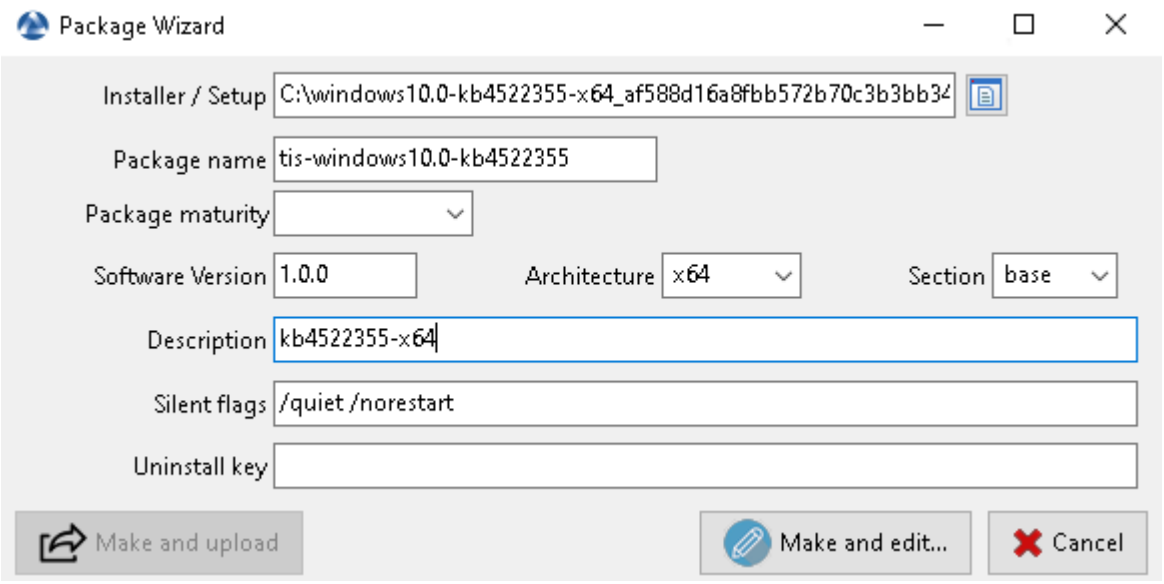


FIG. 6 – Informations requises pour la création du paquet MSU

51.9 Packager des paquets linux simples

Avant de commencer, nous supposons plusieurs conditions :
— You have a graphical interface on your Linux system that you use for developing and testing packages.
— You have installed the **vscode** package from the Tranquil IT repository.
— Your user is named *linuxuser* and is a member of the *sudoers* group.

51.9.1 Créer un modèle de paquet base depuis votre poste linux

— Start up a Command Line utility.
— As *linuxuser*, create a WAPT package template.

```
wapt-get make-template <template_name>
```

Avertissement : Do not launch this command as root or with a 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 : All packages are stored in linuxuser's home (home of the currently logged in user).

VSCode se charge et ouvre le projet de paquet.

— Check the `control` file content.
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.

— Original `control` file.

```
package           : tis-vlc
version           : 0-0
```

(suite sur la page suivante)

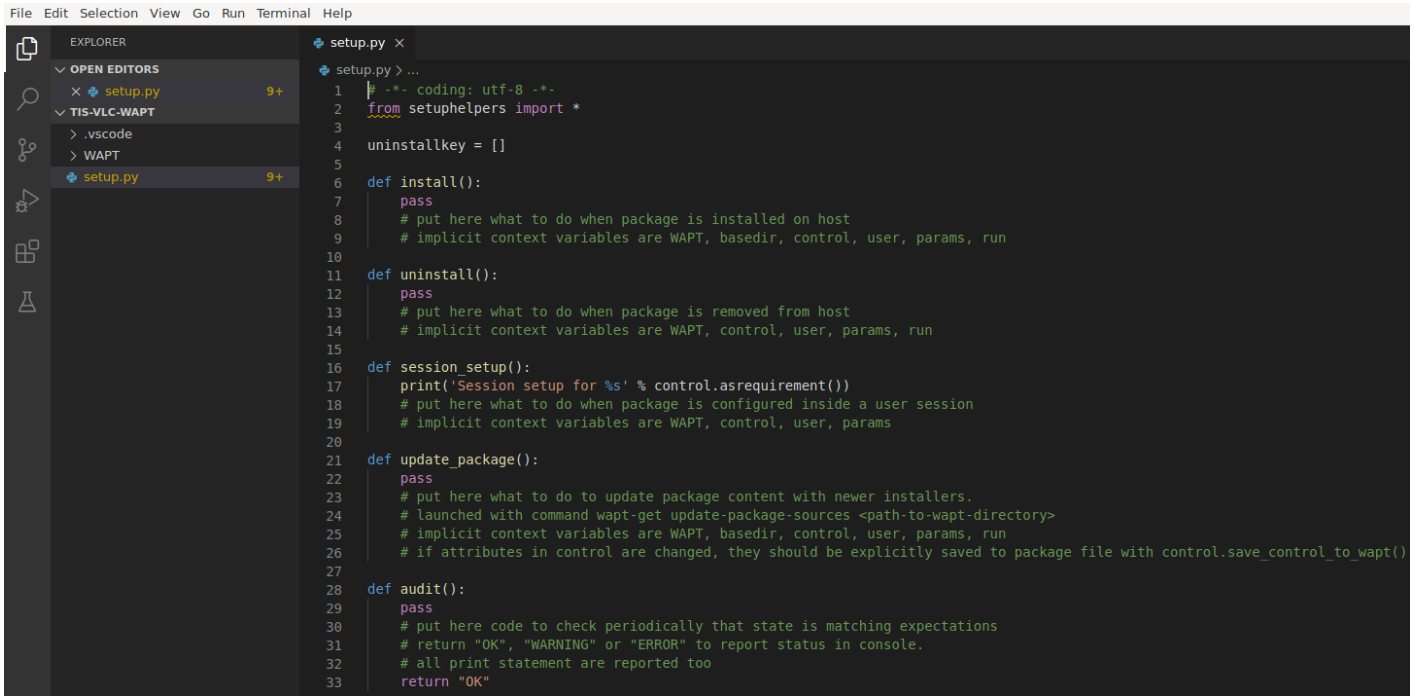


FIG. 7 – Ouverture du VSCode avec le focus sur le fichier *setup*

(suite de la page précédente)

```
architecture      : all
section           : base
priority          : optional
maintainer        : user
description       : automatic package for vlc
```

— Modified `control` file.

```
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.

— Make changes to the code in the `setup.py` file accordingly.

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

uninstallkey = []

def install():
    apt_install('vlc')
```

— Save the package.

51.9.2 Gérer la désinstallation

— Make changes to the `setup.py` file with an uninstall.

```
def uninstall():
apt_remove('vlc')
```

— Launch a *remove* from VSCode *Run Configurations*.

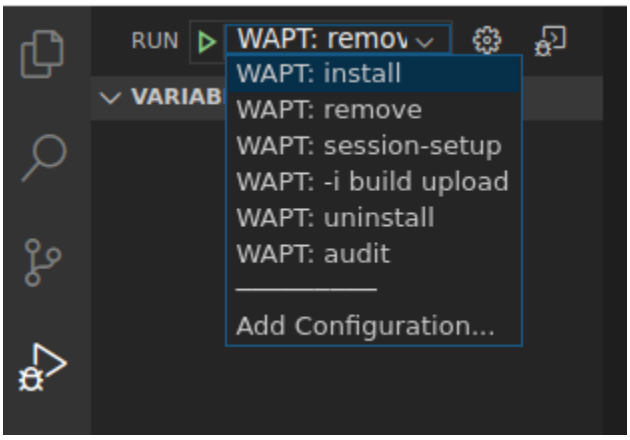


FIG. 8 – À l'issue de la désinstallation, le programme est désinstallé

— Check that the software has been correctly removed.

```
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.

51.9.3 Gérer le session-setup

— Make changes to the `setup.py` file with a `session-setup`;
Dans cet exemple, nous allons créer un fichier :fichier : *vlcrc* par défaut dans le profil de l'utilisateur.

```
def session_setup():
  vlcrc_content="""[qt] # Qt interface
qt-notification=0
qt-privacy-ask=0
metadata-network-access=0
"""

  vlcdir = os.path.join(os.environ['HOME'], '.config', 'vlc')
  path_vlrc = makepath(vlcdir,'vlcrc')
  ensure_dir(vlcdir)
  if not isfile(path_vlrc):
      with open(makepath(vlcdir,'vlcrc')) as f:
          f.write(vlcrc_content)
```

— Launch a *session-setup* from VSCode *Run Configurations*.

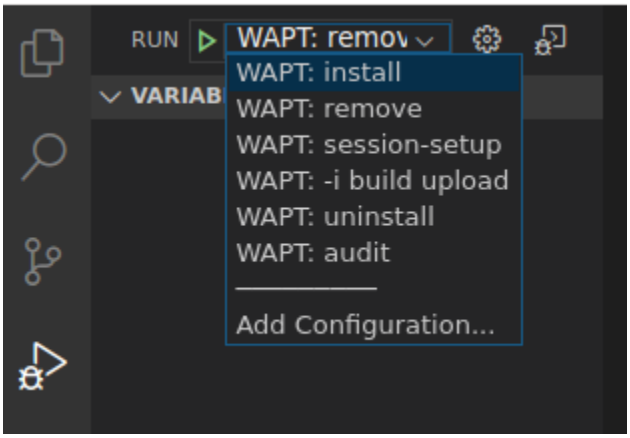


FIG. 9 – À l'issue de la désinstallation, le programme est désinstallé

51.9.4 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*.

51.10 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**.

51.10.1 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 génant 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.** ! !

51.10.2 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`.

- The certificate is sent to the server with the inventory when the WAPT client is first registered.
- The private key is kept by the agent and is only readable locally by *Local Administrators*.

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 privé.

Avec ce mode de fonctionnement le serveur WAPT et les dépôts secondaires n'ont pas connaissance de la donnée sensible.

51.10.3 Cas pratique

Vous trouverez ici un exemple de paquet WAPT ou nous chiffrons un texte dans dans une fonction **`update_package`** puis nous déchiffrons ce texte dans la partie **`install`**.With this mode of operation, the WAPT server and secondary repositories have no knowledge of the sensitive data.

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.You will find here an example of a WAPT package where we encrypt a string of text in an **`update_package`** function and then decrypt this text in the **`install`** function.

Le texte chiffré pour chaque machine est ensuite stocké dans un fichier `encrypt-txt.json` à la racine du paquet.In this example, the **`update_package`** function allows us to browse the WAPT server database to retrieve the certificate from each machine and then encrypt the sensitive text with it.

Lors de l'installation du paquet, l'agent WAPT prendra le texte chiffré et le déchiffrera avec sa clé privé.The encrypted text for each machine is then stored in a `encrypt-txt.json` file at the root of the package.

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

(suite sur la page suivante)

(suite de la page précédente)

```
from waptcrypto import SSLCertificate
import waptguihelper
import base64

def install():
    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(crt_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 text déchiffré avec un :command :`print` lors de l'installation**.

CHAPITRE 52

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 créerez 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.

52.1 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%waptconsolewaptconsole.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	**`vscode`** ou **`code`**
Microsoft Visual Studio Codium	**`vscodium`** ou **`codium`**

Exemple de configuration de `waptconsole.ini` :

```
[global]
...
editor_for_packages=vscode
```

52.2 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) :

- **`vscodium`**;
- **`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	**`vscode`** ou **`code`**
Microsoft Visual Studio Codium	**`vscodium`** ou **`codium`**
Nano	**`nano`**
Vim	**`vim`**
Vi	**`vi`**

```
[global]
...
editor_for_packages=vim
```

CHAPITRE 53

Configurer WAPT pour utiliser un éditeur de code personnalisé

53.1 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 *setup_filename*
PyCharm	C:\Program Files\JetBrains\PyCharm Community Edition 2019.3.2\bin\pycharm64.exe *wapt_sources_dir*

```
[global]
...
editor_for_packages=C:\Program Files\Notepad++\notepad++.exe {setup_filename}
```

53.2 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 *wapt_sources_dir*

```
[global]
...
editor_for_packages=/opt/pycharm/bin/pycharm_x64 {wapt_sources_dir}
```

53.2.1 Arguments personnalisés

Tableau 3 – Les arguments peuvent être passés dans la commande editor_for_packages

Argument	Description
{setup_filename}	Lance l'éditeur de code personnalisé et édite le fichier WAPT setup.py
{control_filename}	Lance l'éditeur de code personnalisé et modifie le fichier control des paquets WAPT
{wapt_sources_dir}	Lance l'éditeur de code personnalisé et ouvre le dossier du paquet WAPT
{wapt_base_dir}	Lance l'éditeur de code personnalisé et ouvre le dossier d'installation WAPT

This page details various windows of the WAPT console.

CHAPITRE 54

Menus

Detailed tabs of the menu bar

54.1 Onglet Fichier (File)

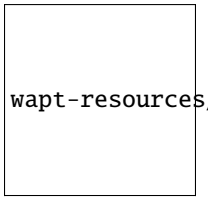


FIG. 1 – Exit the WAPT console

The action closes the WAPT console. The keyboard shortcut is `Ctrl+Q`.

54.2 Onglet Affichage (View)

54.2.1 Rafraîchir

The action refreshes the display of the WAPT console.

The keyboard shortcut is `F5`.

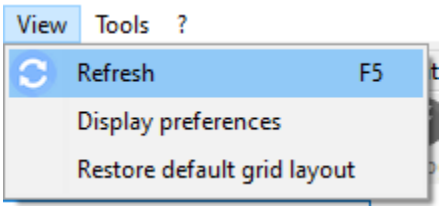


Fig. 2 – Menu option for refreshing the display of the WAPT console

54.2.2 Afficher les préférences

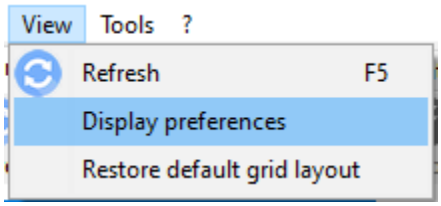


Fig. 3 – Menu option for setting display preferences for the WAPT console

This window allows you to set your own preferences for the WAPT console.

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 courrant	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

54.2.3 Restaurer l'affichage de la grille par défaut

Re-apply the default layout for the WAPT console. For example, you can remove the column *Hardware inventory added in this section of the documentation*.

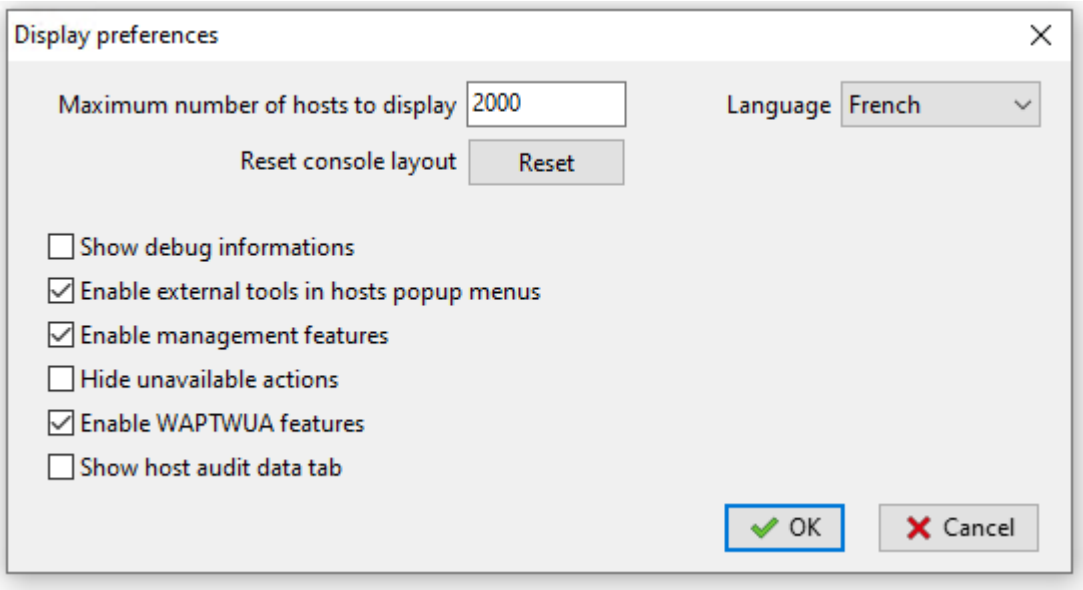


FIG. 4 – Window for personalizing the WAPT console

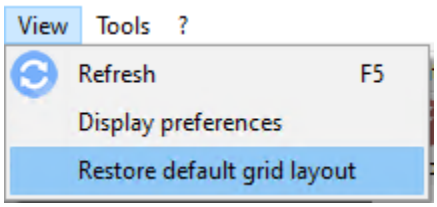


FIG. 5 – Menu option for restoring the default grid layout of the WAPT console

54.3 Onglet outils

54.3.1 Changer le mot de passe Superadmin du serveur WAPT

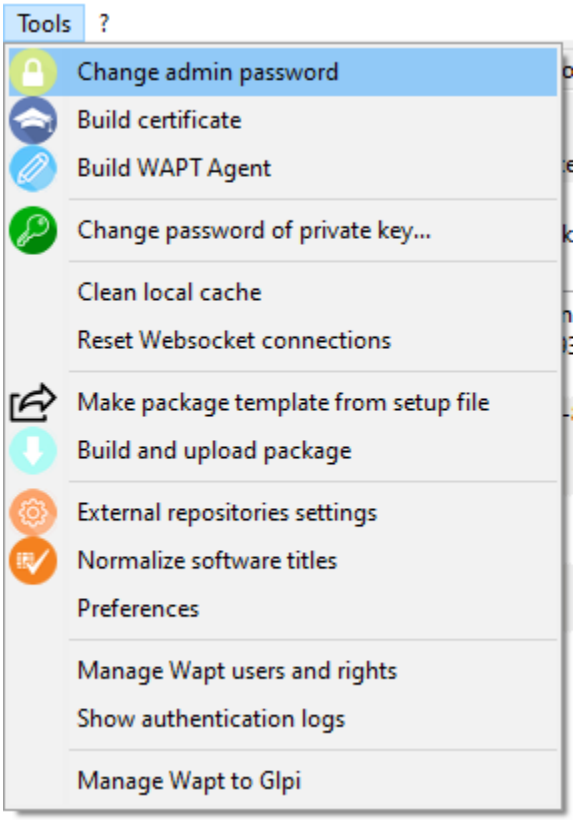


FIG. 6 – Menu option for changing the Superadmin password of the WAPT Server

This window allows you to update the Super Admin password initially created during the postconfiguration of the WAPT Server.

To change the WAPT Server password, fill in the old password and enter a new one.

54.3.2 Construire un certificat Administrateur

Référez-vous à la documentation *générer le certificat Administrateur pour signer des paquets WAPT*.

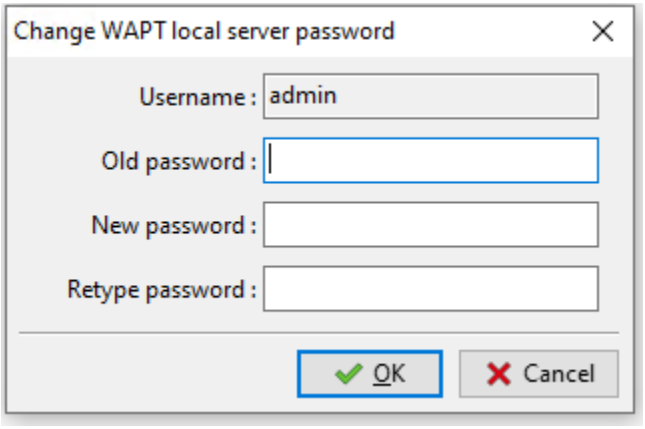


FIG. 7 – Window for changing the Superadmin password of the WAPT Server

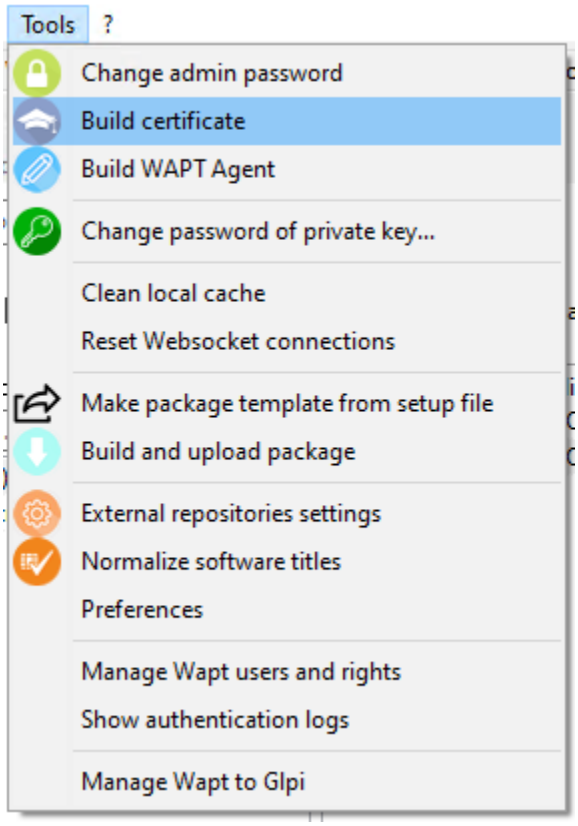


FIG. 8 – Menu option for creating a certificate for a WAPT Administrator

54.3.3 Construire l'agent WAPT

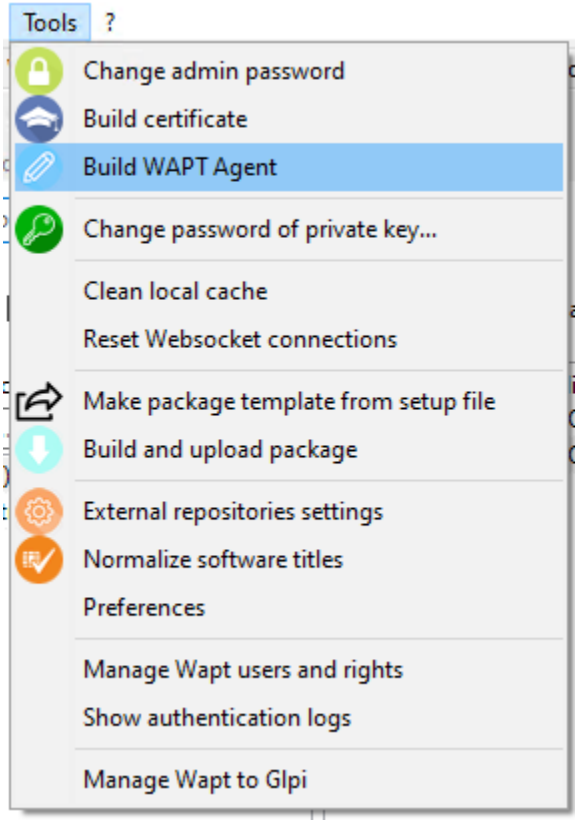


FIG. 9 – Menu option for building the WAPT agent

Référez-vous à la documentation *construire l'installeur de l'agent WAPT*.

54.3.4 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*.

To change the private key password, fill in the old password and enter a new one.

54.3.5 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 *Tools* → *Clean local cache*.

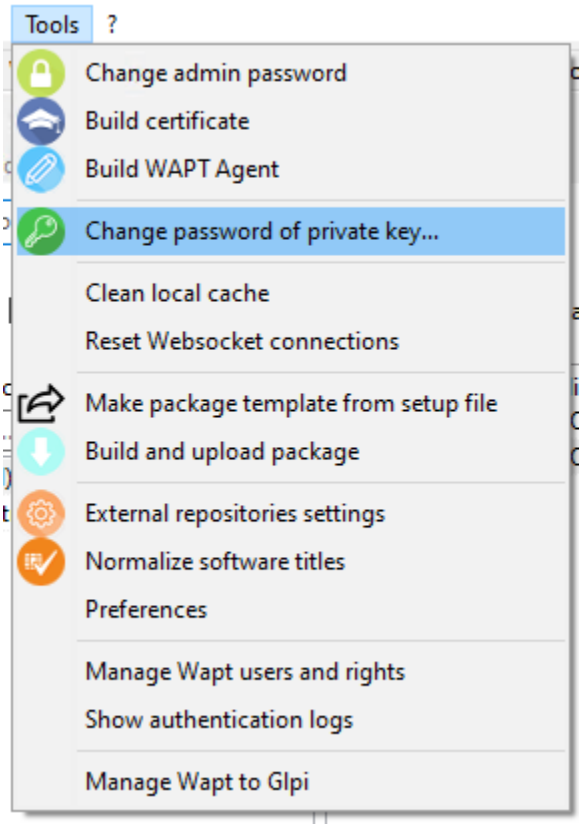


FIG. 10 – Menu option for changing the password of the Administrator's private key

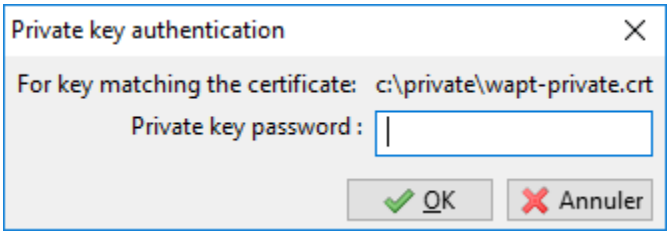


FIG. 11 – Window for entering the password to unlock the private key

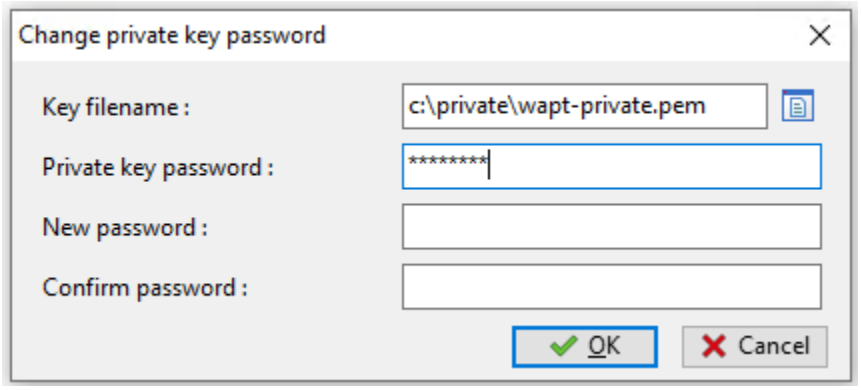


FIG. 12 – Window for changing the password of the Administrator's private key

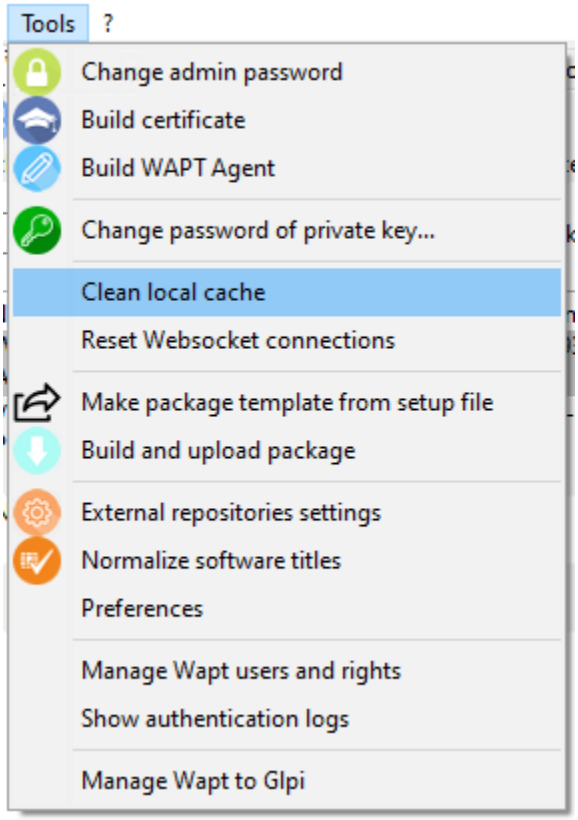


FIG. 13 – Menu option for cleaning up the local cache

54.3.6 Réinitialiser les connexions Websocket

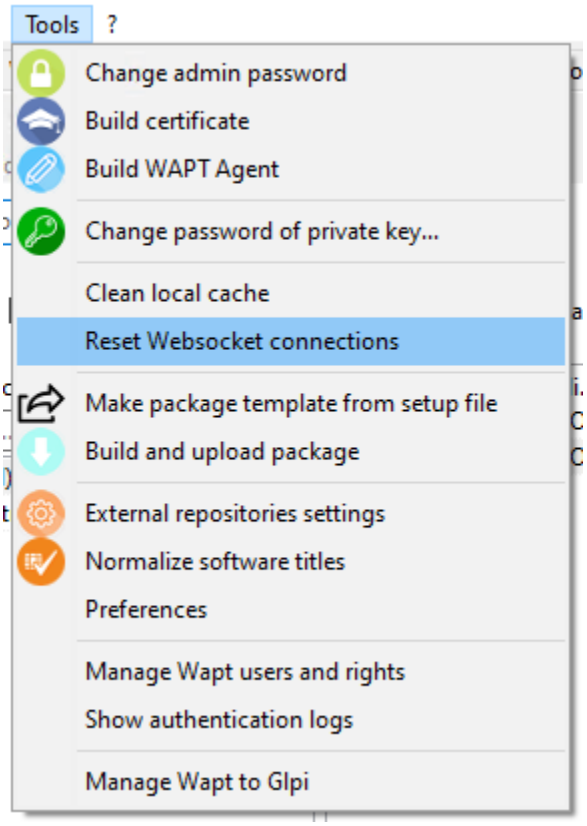


FIG. 14 – Menu option for resetting Websocket connections

Utilisez cette méthode si vous avez redémarré le service waptserver sans avoir redémarré le service web **`Nginx`**.

54.3.7 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*.

54.3.8 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*.

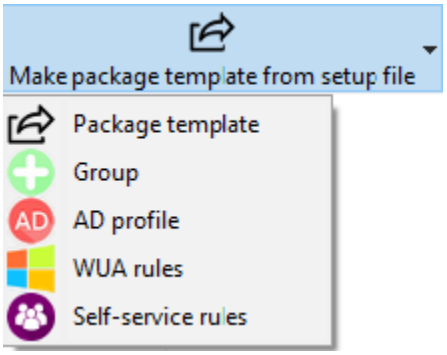


FIG. 15 – Menu option for making a WAPT package template from a setup file

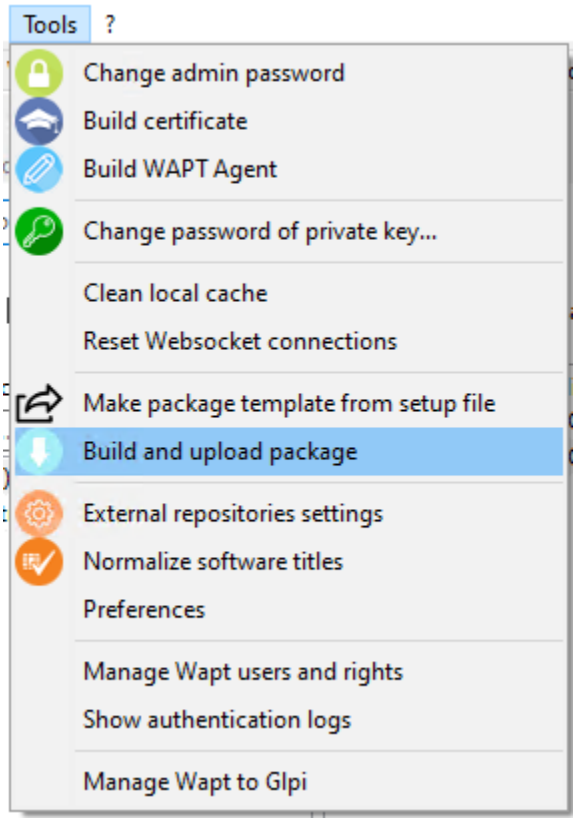


FIG. 16 – Menu option for building and uploading a WAPT package from the WAPT console to a repository

54.3.9 Paramètres des dépôts externes

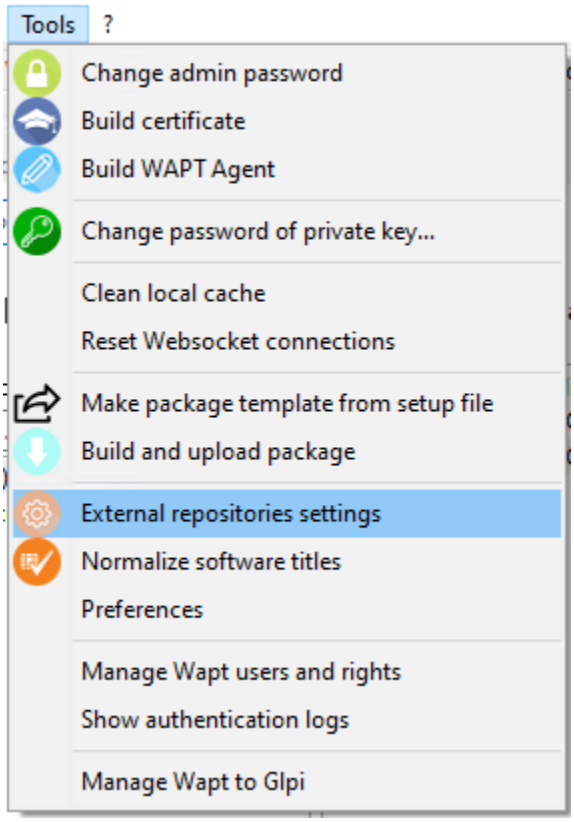


FIG. 17 – Menu option for setting external repositories

You can add external repositories to get packages from different sources.

TABLEAU 2 – Liste des actions disponibles applicables aux dépôts

Nom	Description	Exemple
Nom du dépôt	Nom du dépot dans la liste	wapt-template
Inscrire un nouveau dépôt	Effacer les certificats	
Désinscrire le dépôt	Delete the selcted repository	
Dépôt de paquets externe	adresse du dépot	https://store.wapt.fr/wapt
Rechercher les certificats	Download the certificate of the repository	
proxy http à utiliser (si besoin)	adresse du proxy	http://proxy.mydomain.lan
Paramètres avancés	Affichage des paramètres avancés	Coché
Vérifier le certificat HTTPS du serveur	Utilisation des certificats HTTPS	Coché

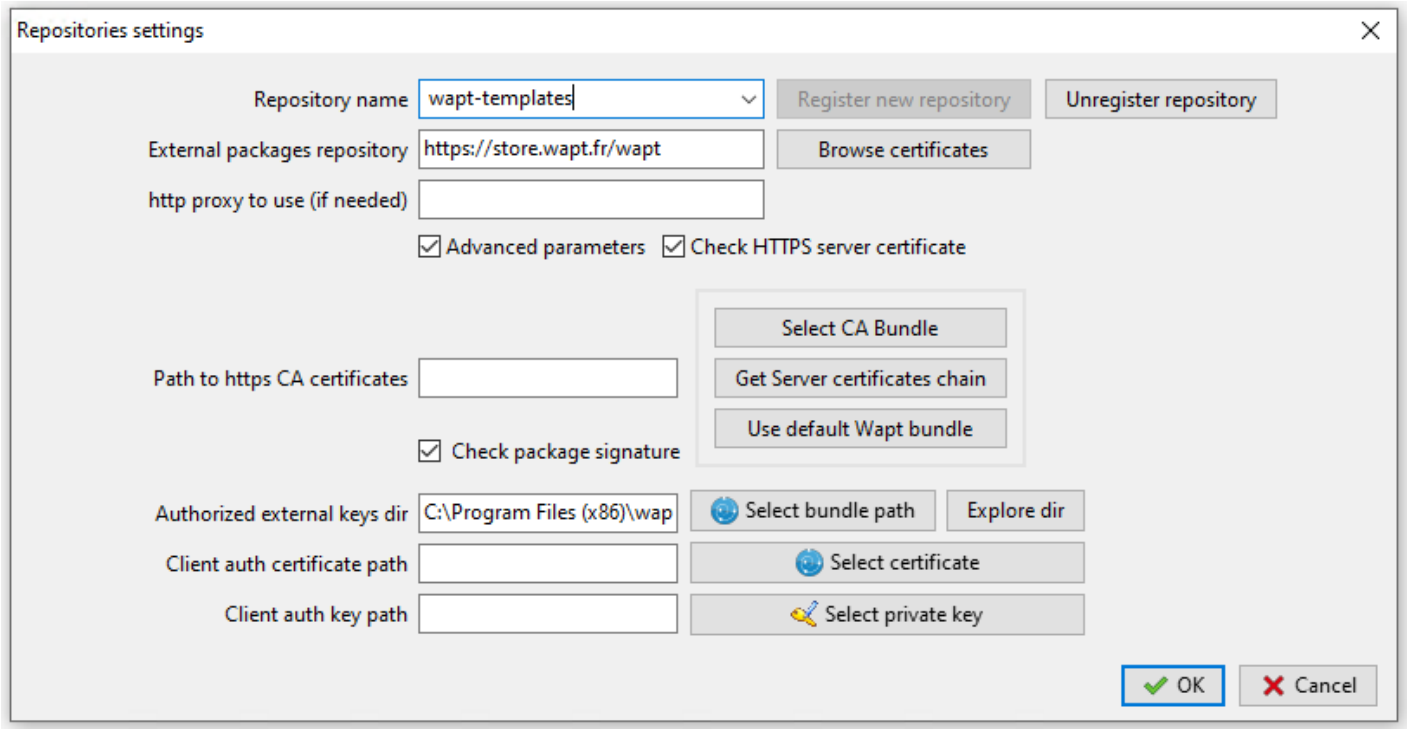


FIG. 18 – Window for setting up external repositories

54.3.10 Normaliser les noms de logiciels

Pour plus d'informations, référez-vous à la documentation *normaliser le nom des logiciels*.

54.3.11 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 ;

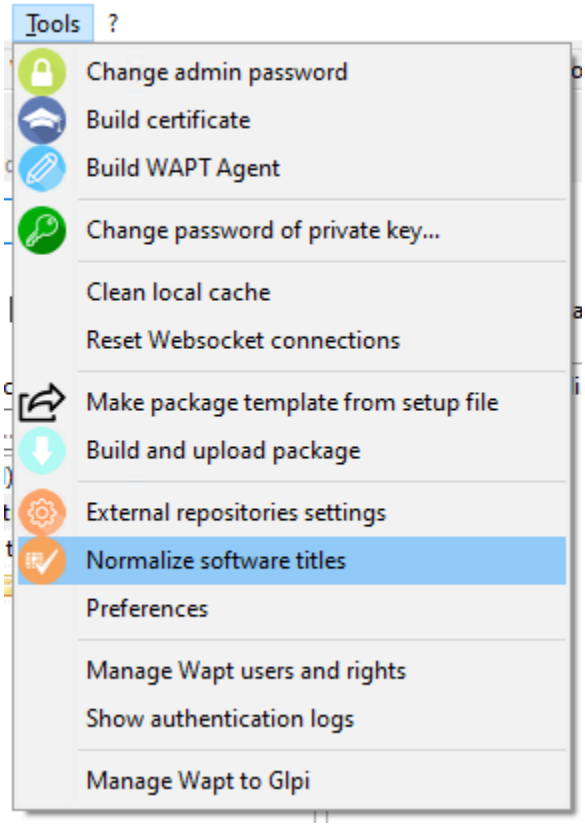


FIG. 19 – Menu option for normalizing software titles

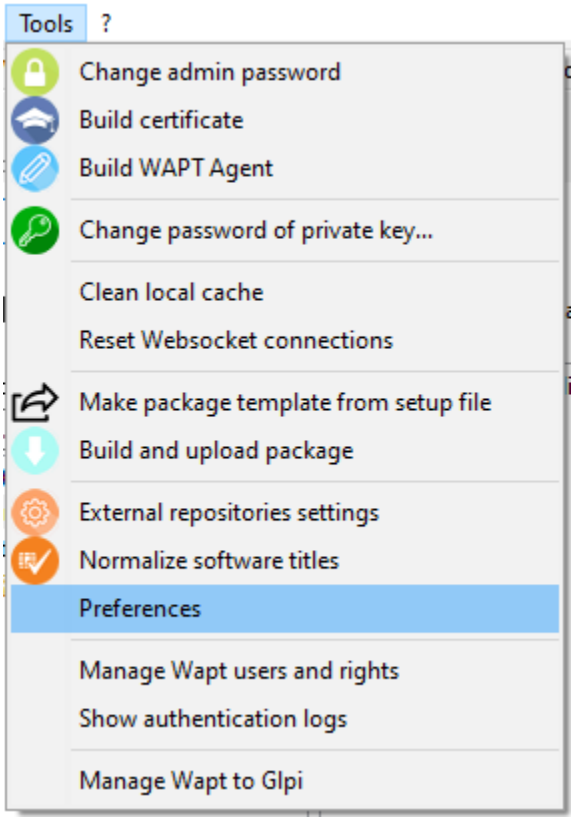


FIG. 20 – Menu option for configuring the WAPT console preferences

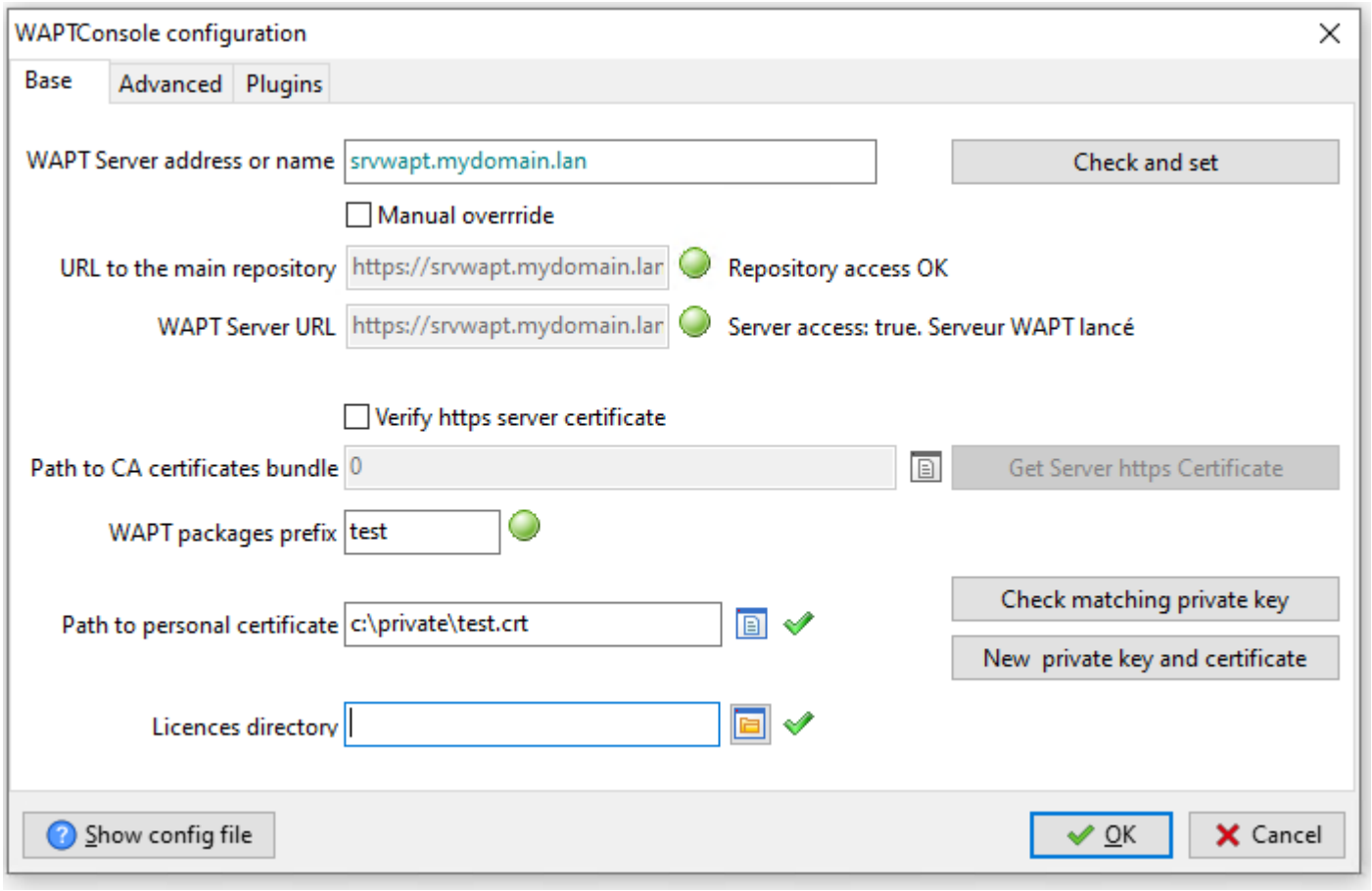


FIG. 21 – Window for the basic configuration of the WAPT console

Arguments	Description	Exemple
L'adresse ou le nom du serveur WAPT	URL (IP or FQDN) of the WAPT Server	*srvwapt.mydomain.lan*.
Vérifier	Check whether the server exists and set this configuration. Write the WAPT Server URL to `waptconsole.ini`	
Spécifier manuellement	Used for bypass automatic generation.	Coché
URL du dépôt principal WAPT	Set repository address of WAPT (only if *Manual override* is checked).	http://srvwapt.mydomain.lan/wapt/
URL du serveur WAPT	Set the WAPT Server address (only if *Specify manually* is checked).	*https ://srvwapt.mydomain.lan/*
Vérifie le certificat HTTS du serveur	Indicates whether the HTTPS certificate must be verified.	Coché
Chemin vers le bindle de certificats CA	Path to the CA bundle of certificates that will allow certificates to be verified. Visit *the documentation on activating HTTPS verification*.	`C:\Program Files (x86)\wapt\lib\site-packages\certifi\cacert.pem`
Récupération certificat serveur https	Retrieve the https certificate chain from the WAPT Server.	
Préfixe des paquets WAPT	Le préfixe donné aux paquets lors des réplications.	tis
Chemin du certificat personnel	Path to the certificate associated with the private key used to sign packages.	`C:\private\mykey.crt`
Répertoire des licences	Chemin vers les licences en version Enterprise. Configuré (vide) par défaut dans le dossier d'installation de WAPT. Il est possible de renseigner un autre chemin.	`X:\licence`
Check matching private key	Check whether the `.crt` certificate matches the `.pem` private key.	
New private key and certificate	Create a new *private key / certificate* pair.	
Afficher le fichier de configuration	Open `%appdata%\Local\waptconsole\waptconsole.ini` 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 sure que vous vous connectez au bon serveur.

Configuration avancé

Onglet Avancé pour les options de configuration avancé ;

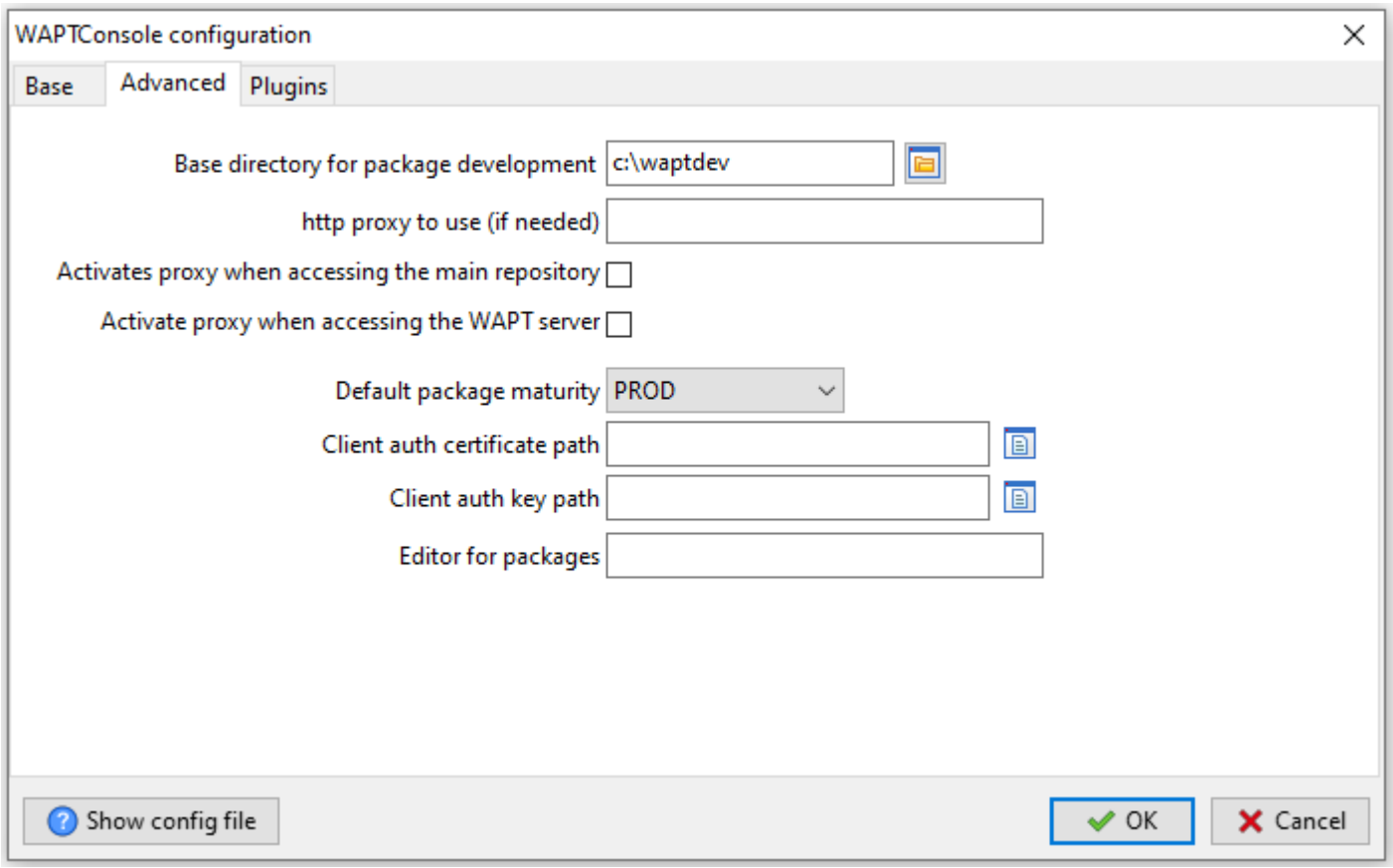


FIG. 22 – Window for the advanced configuration of the WAPT console

Arguments	Description	Exemple
Répertoire de base pour le développement de paquet	Indicates the path to the directory for storing packages being developed.	`C:\waptdev`
proxy http à utiliser (si besoin)	Indicates a proxy server to be used by the WAPT console when accessing the WAPT repository or the WAPT Server.	http://srvproxy.mydomain.lan:8080
Active le proxy pour l'accès au dépôt principal	Activate proxy settings for connecting to the WAPT repository.	Coché
Active le proxy pour accéder au serveur WAPT	Use the proxy when accessing the WAPT server.	Coché
Maturité des paquets par défaut	Default maturity for imported packages.	PROD
Chemin certificat SSL client	Path to the certificate for using *Client side SSL authentication*.	Vide
Chemin clé SSL client	Path to the key for using *Client side SSL authentication*.	Vide
Editeur pour les paquets	Default development environment for importing WAPT packages.	PyScripter
Afficher le fichier de configuration	Ouvre le fichier `waptconsole.ini` dans `%appdata%\Local\waptconsole`	Tous les paramètres de la console WAPT

Outils externes

Pluggins allow you to add custom actions not natively available with WAPT.

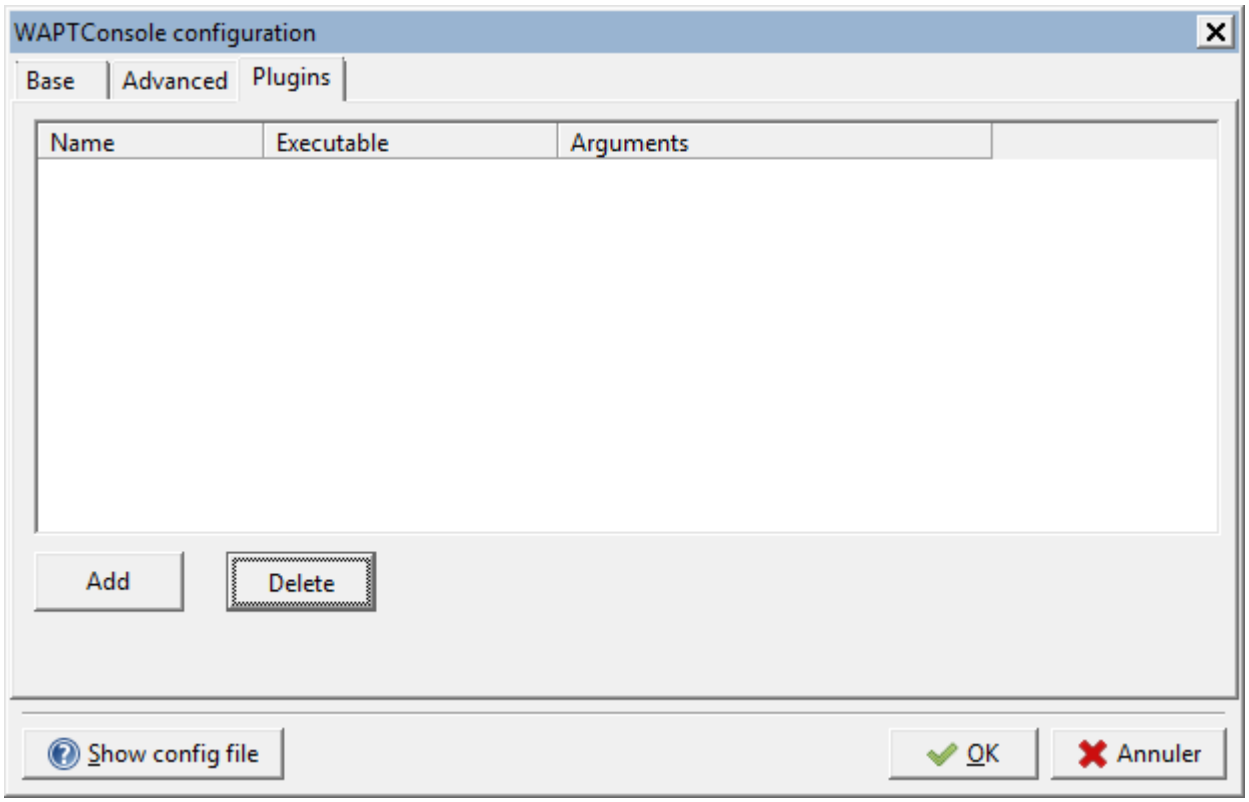


FIG. 23 – Créet un plugin

Click *Add* to add plugins, then edit the corresponding columns.

Colonne	Description
Nom	Name that will appear in the menu.
Exécutable	Path of the executable that will be executed.
Arguments	Arguments passed to the executable. Some variables can be used like {ip}, {uuid} or {computer_fqdn}.
Afficher le fichier de configuration	Open `waptconsole.ini` file in `%appdata%/Local/waptconsole`.

Les plugins vont alors apparaître dans le menu :

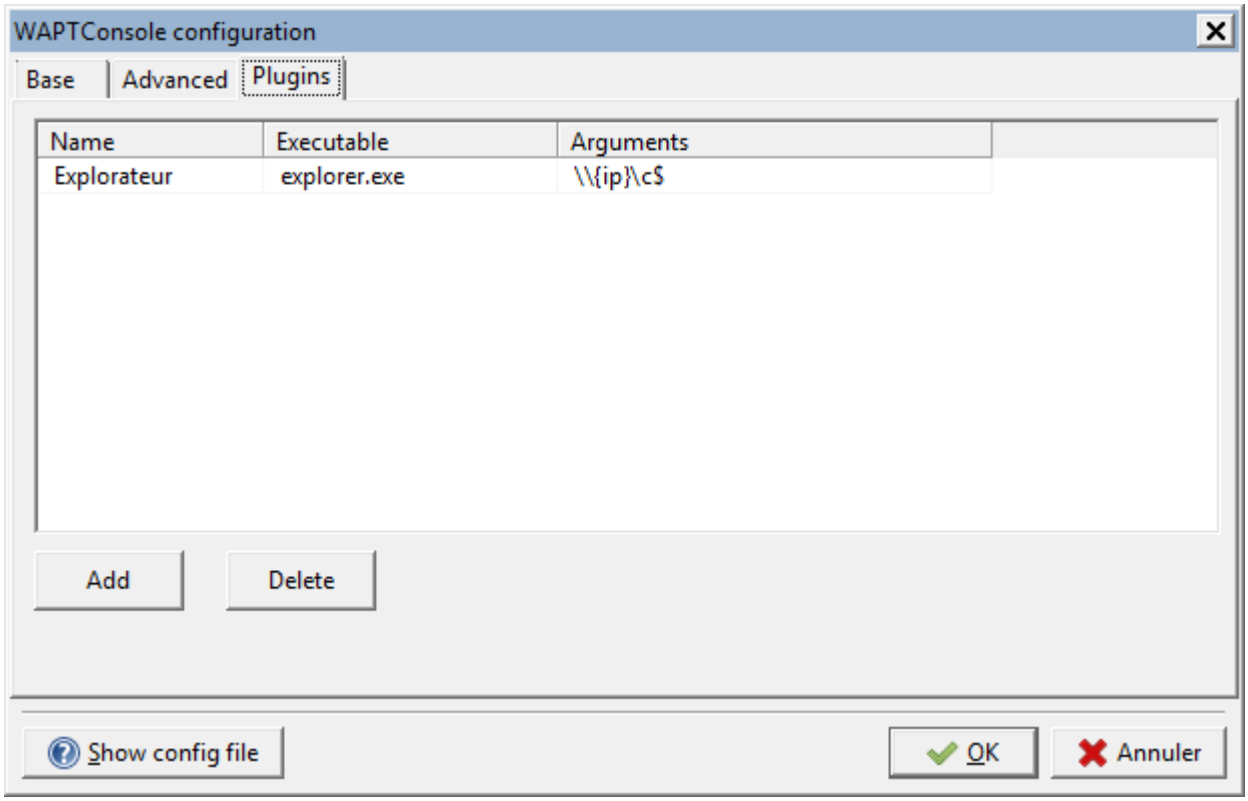


Fig. 24 – Inserting « Explorer » as a plugin with IP variables

54.4 L'onglet ?

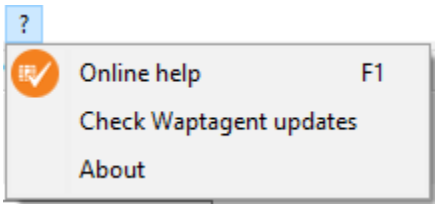


FIG. 25 – More info tab

54.4.1 Aide en ligne

Online help opens a web browser with this documentation.

54.4.2 Vérifier si l'agent WAPT est à jour

Check Waptagent updates checks whether an agent update is available on the server.

54.4.3 A propos

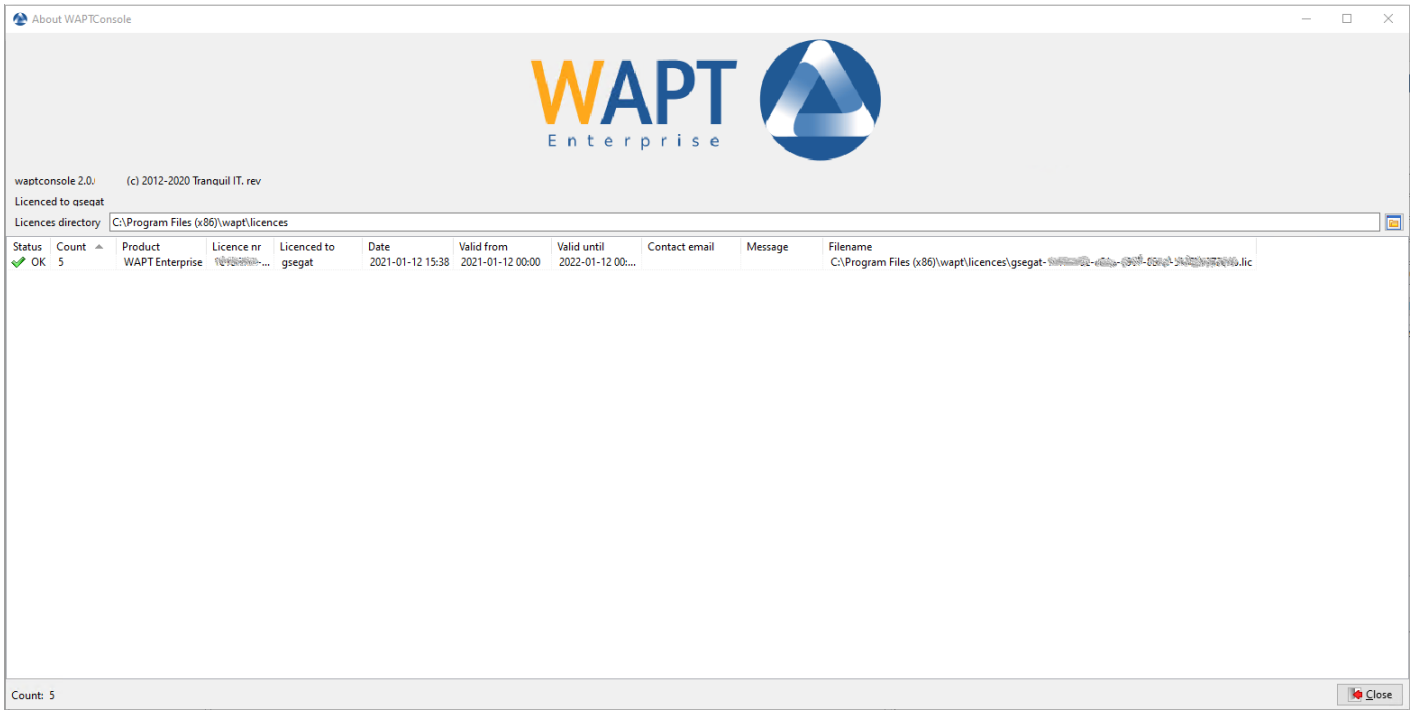


FIG. 26 – About tab

Tableau 3 – Liste des actions disponibles

Colonne	Description	Exemple
Status	Statut de validité de la licence	*Ok*
Nombre	Nombre de poste compris dans la licence	*Fermer*
No de licence	Numéro de licence	`12345678-aaa-bbb-ccc-123456789acb`
Concédé à	Détenteur de la licence	pdegalles
Date	License creation date	2021-01-12 00 :00
Début validité	Start date for the licence.	2021-01-12 00 :00
Fin de validité	End date for the licence.	2022-01-12 00 :00
Email de contact	Email of the licence's owner.	pdegalles@paysdegalles.org
Message	Message displayed when license has an *ERROR* status.	*SSLVerifyException : SSL signature verification failed for certificate*
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`

CHAPITRE 55

Onglet inventaire

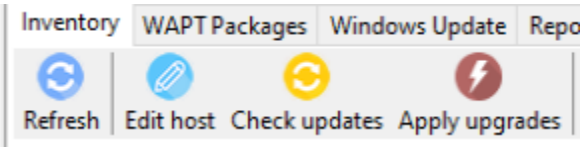


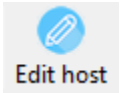
FIG. 1 – Onglet inventaire

55.1 Rafraîchir

Refresh refreshes the WAPT console display.

The keyboard shortcut is `F5`.

55.2 Modifier la machine

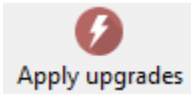


Edit host allows you to *edit* host package.

55.3 Vérification des mises à jours

Check updates allows you to *check updates* for the selected host(s).

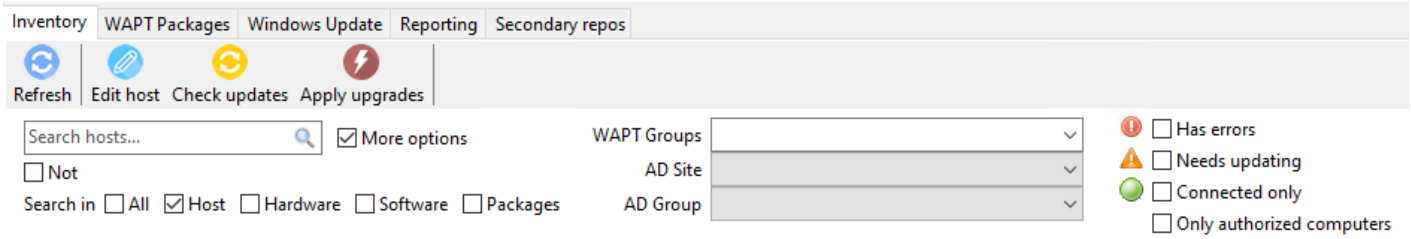
55.4 Lancer les installations



Apply updates allows you to *apply upgrades* on the selected host(s).

CHAPITRE 56

Effectuer une recherche globale sur tous les postes



The filters allow you to search hosts based on *multiple criterias*.

CHAPITRE 57

Les OU AD dans WAPT

WAPT console is not directly connected to Active Directory. The arborescence is built from WAPT agent inventory information.

If the tree structure is not up-to-date, you have to launch a *Check Updates* on a *client* or on *OU « .*

57.1 Inclure les postes des sous-dossiers

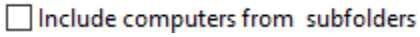


FIG. 1 – Including computers from subfolders

This option allows you to display all available hosts in OU folders and subfolders.

If not checked, only available hosts in selected OU will be displayed in the console.

57.2 Searching Organizational Units

FIG. 2 – Searching an Organizational Unit

This option allows you to search in OU folders.

Indication : Les filtres fonctionnent avec les expressions régulières.

57.3 Displaying Organizational Units

Fig. 3 – Unfolding Organizational Unit folders

Unfolding Organizational Unit folders allows you to display nested OU.

57.4 Creating or editing Organizational Unit package

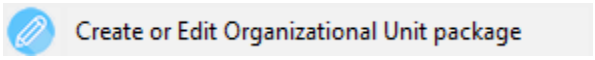


Fig. 4 – Menu option for creating or editing Unit packages

This option allows you to *edit or create Unit package*.

57.5 Checking for updates on all host of an Organizational Unit

Fig. 5 – Menu option for checking updates for hosts in OU

This option allows you to *check updates* of all the hosts in the OU.

57.6 Applying upgrades on all hosts in an Organizational Unit

This option allows you to *apply upgrades* on all hosts in the OU.

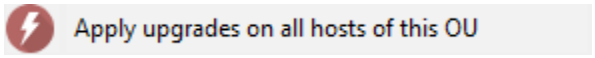


FIG. 6 – Menu option for applying upgrades on hosts in a OU

CHAPITRE 58

Inventaire de l'hôte

FIG. 1 – Inventaire de l'hôte

Main inventory displays every host according to the selected filters.

58.1 Statut de l'hôte

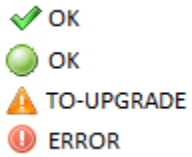


FIG. 2 – Statut de l'hôte

Les différentes colonnes peuvent avoir différents statut.

Status	Description
OK	Host does not report issue.
OK	Host is connected.
TO-UPGRADE	Host has a pending install / update.
ERROR	Host has errors.

58.2 Sous onglet de l'hôte

Some columns are not displayed by default, you can select more columns with a right-click on the inventory header.

Status	Description
Status (host_status)	*Status* of the host.
Joignable (reachable)	If the host is seen by the WAPT server.
Audit status (audit_status)	Audit status of all packages on the host.
WUA (waptwua_status.status)	*WUA service* status via WAPT.
Host (computer_fqdn)	FQDN of the host.
Description (description)	Description of the host
Manufacturer (manufacturer)	Manufacturer of the host
IP addresses (connected_ips)	IP address list of the host
Plateform (plateform)	Architecture du processeur
Operating system (os_name)	Nom du système d'exploitation
Model (productname)	Model for the host
Numéro de série (serialnr)	Host serial number
Last seen on (last_seen_on)	Dernière remonté d'inventaire de la machine
Logged in users (last_seen_on)	Current logged in user
MAC address (mac_addresses)	Adresse MAC correspondantes au IP des machines
WAPT version (wapt_version)	WAPT version deployed on the host
uuid (uuid)	Identifiant UUID de la machine
purchase_order (purchase_order)	Numéro de commande d'achat de la machine
purchase_date (purchase_date)	Date d'achat de la machine
Domain controller (host_info/domain_controller)	Host Domain controller
Groups (depends)	Groups which the host is a member of
Domaine (host_info/domain_controller_address)	Nom de domaine
Listening address (listening_address)	Adresse d'écoute
Domain contoller address (host_info/domain_controller_address)	Adresse du contrôleur de domaine
AD DN	Distinguished Name de la machine
Last logged on user (last_logged_on_user)	Dernier utilisateur connecté
Registration Method (registration_auth_user)	User that has registered the host with the WAPT Server
Repositories (repositories)	Repositories seen by the host

You can add columns from *Software Inventory* by using a drag and drop.

You have to refresh the WAPT console (`F5`) to see the additional data.

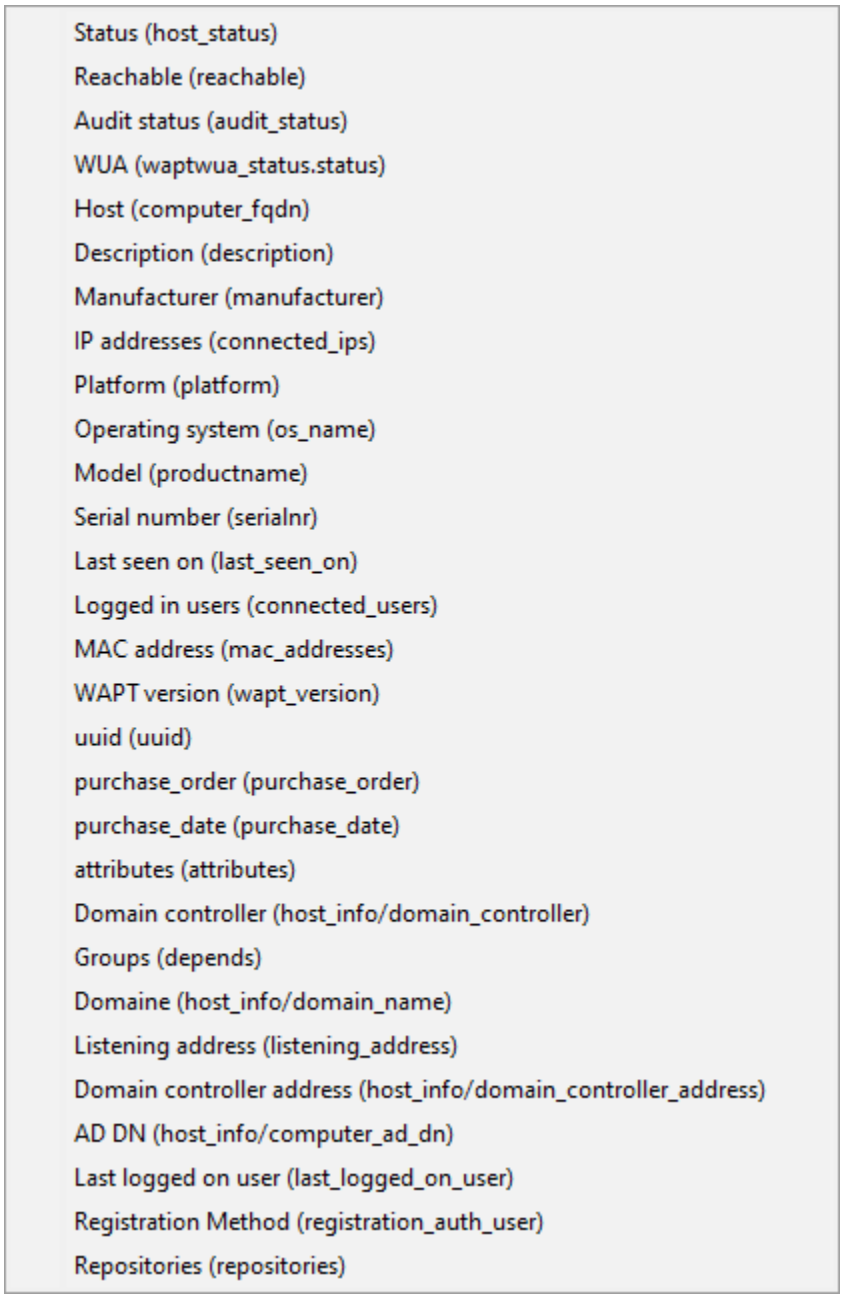


FIG. 3 – Host inventory layout

FIG. 4 – Dragging and dropping inventory items in the WAPT console grid

58.3 Actions détaillées sur les hôtes

58.3.1 Menu de configuration de l'hôte

The menu options are accessible using Right-Click on the selected host(s).

Modifier la machine

The *Edit host* button allows you to *edit* the configuration on the selected host(s).

Vérification des mises à jours

The *Check updates* button allows you to *check for available updates* for the selected host(s).

Lancer les installations

The *Apply upgrades* menu option allows you to *trigger upgrades* on the selected host(s).

Lancer les installations en attentes pour les applications non lancées

The *Apply upgrades for not running applications* menu option allows you to *apply upgrades* on the selected host(s) **if the software is not being currently used by a user**.

Proposer la mise à jour aux utilisateurs

The *Propose Upgrades to logged on users* menu option allows you to propose upgrades to currently logged-in users of the selected host(s). **A popup will then appear on the user's terminal allowing the user to trigger or postpone the offered upgrade**.

This menu option is equivalent to *waptexit*.

Envoyer un message aux utilisateurs

The *Send a message to users* menu option allows you to send a message to logged-in users of the selected host(s).

Lancer l'audit des paquets

The *Run packages audit* menu option allows you to trigger manually a *compliance audit* on the selected host(s).

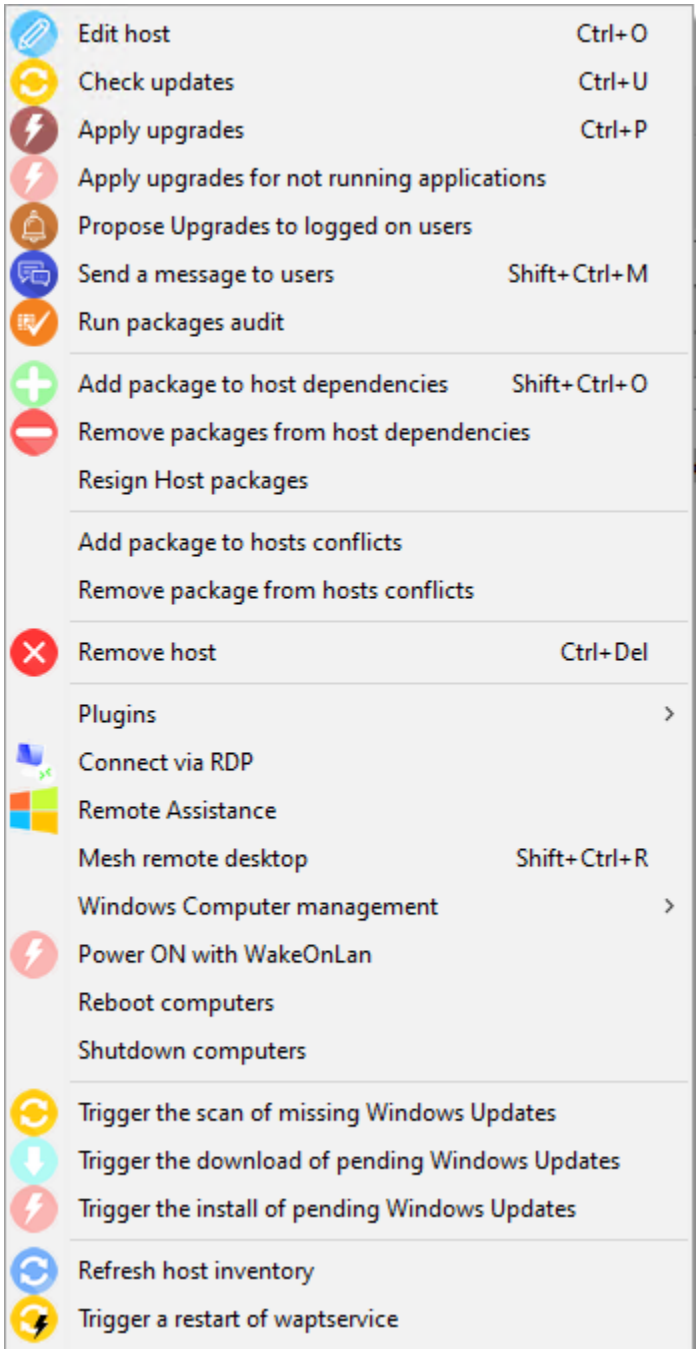


FIG. 5 – Menu de configuration de l'hôte

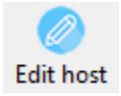


FIG. 6 – Edit host button

FIG. 7 – Check updates button

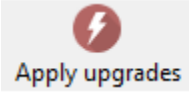


FIG. 8 – Menu option for triggering upgrades

Add package(s) to host(s)

The *Add packages to host* menu option allows you to *add a list of WAPT packages* on the selected host(s).

Remove package(s) from host(s)

The *Remove package(s) from host(s)* menu option allows you *to remove a list of WAPT packages* from the selected host(s).

Re-sign host packages

The *Resign host packages* menu option allows you to *re-sign the WAPT configuration* of the selected host(s).

À faire : continue from here (vcardon)

58.3.2 Ajouter un paquet en conflit avec l'hôte

The *Add package to host conflicts* menu option allows you to forbid a package from being installed on the selected host(s).

Autoriser des paquets

The *Remove package from host conflicts* menu option allows you to re-authorize a forbidden package on the selected host(s).

FIG. 9 – Menu option for triggering upgrades on software not in use

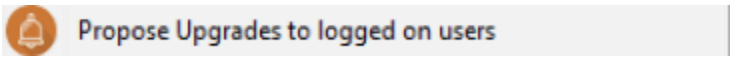


Fig. 10 – Menu option for proposing Upgrades to logged on users

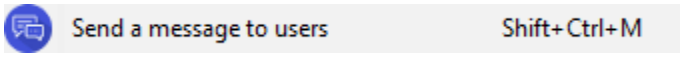


Fig. 11 – Menu option for sending a message to users

Fig. 12 – Menu option for adding package(s) to host(s)

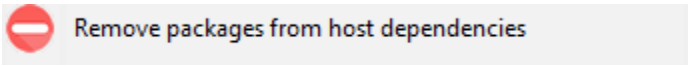


Fig. 13 – Menu option for re-signing a host configuration

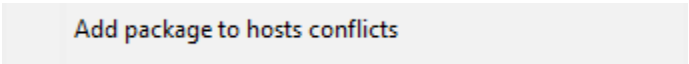


Fig. 14 – Menu option for adding a forbidden package

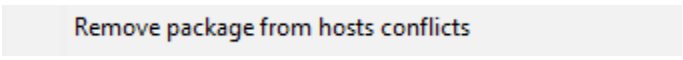


Fig. 15 – Menu option for removing a package from the conflict list

Supprimer le poste

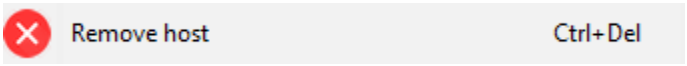


Fig. 16 – Menu option for removing host(s)

The *Remove host* menu option allows you to remove the selected host(s) from the WAPT Server database.

2 options are available :

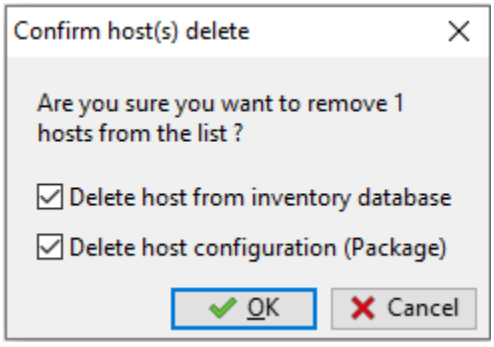
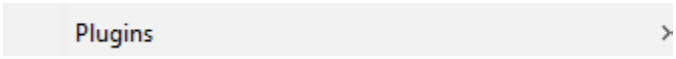


Fig. 17 – Window to confirm removing host(s)

— Delete host from inventory database to delete inventory data.
— Delete host configuration (Package) to delete host package data.

Outils externes



See here.

Se connecter en RDP

The *Connect via RDP* menu option allows you to connect with RDP on the selected host.

Assistance à distance

The *Remote Assistance* menu option allows you to launch Windows remote assistance.

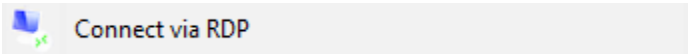


Fig. 18 – Menu option for connecting to the host via RDP

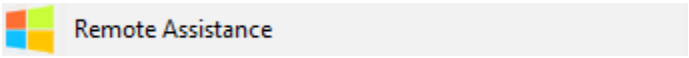


Fig. 19 – Menu option for connecting to the host via Remote Assistance

Bureau à distance Mesh

The *Mesh remote desktop* menu option allows you to use Mesh Commander from the WAPT console.

Gestion Windows de l'ordinateur

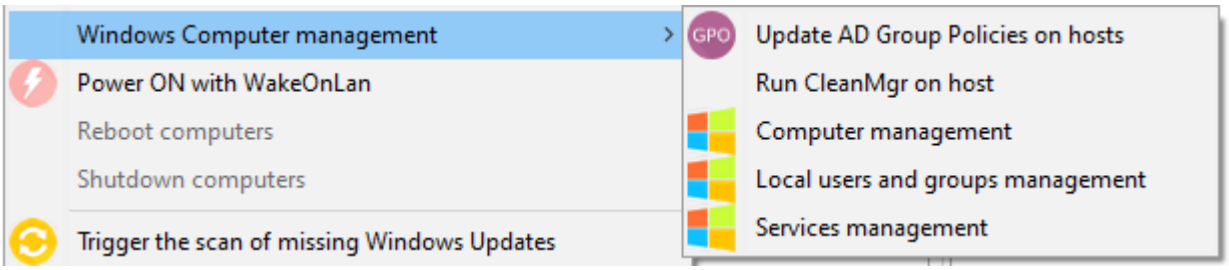


Fig. 20 – Menu option for remotely managing a host

The *Windows Computer management* menu option allows you to launch the Windows remote management tools from the WAPT console.

Force la mise à jour des GPO sur les machines

The *Update AD Group Policies on host* menu option allows you to launch **`gpupdate /force`** command on the selected host(s).

Lance le CleanMgr sur l'hôte

The *Run CleanMgr on host* menu option allows you to remotely trigger the **`cleanmgr`** tool on the selected host(s).

Gérer l'ordinateur

The *Computer management* menu option allows you to remotely launch **`compmgmt.msc`** on the selected host.

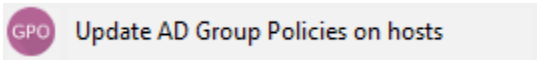


Fig. 21 – Menu option for remotely applying GPO on hosts

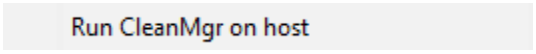


Fig. 22 – Menu option for remotely triggering CleanMgr on hosts

Gérer les utilisateurs et groupes

The *Local users and groups management* menu option allows you to remotely launch **`lusrmgr.exe`** on the selected host.

Gérer les services

The *Services management* menu option allows you to remotely launch **`services.msc`** on the selected host.

Démarre avec le WakeOnLan

The *Power ON with WakeOnLan* menu option allows you to remotely wake selected host(s) if magic packages are authorized.

Redémarrer les machines

The *Reboot computers* menu option allows you to remotely reboot the selected host(s).

This feature is disabled by default, see *Configuring the WAPT agent* for enabling the feature.

Arrêter les machines

The *Shutdown computers* menu option allows you to remotely shut down the selected host(s).

This feature is disabled by default, see *Configuring the WAPT agent* for enabling the feature.

Lancer la recherche des Mises à jour Windows à appliquer

The *Trigger the scan of missing Windows Updates* menu option allows you to remotely trigger the scan for missing Windows Updates on the selected host(s).

For more information, visit this *documentation*.

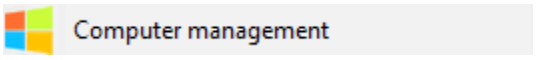


Fig. 23 – Menu option for remotely launching the Computer Management tool on the host

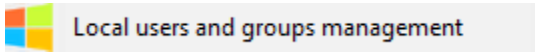


Fig. 24 – Menu option for launching the Local users and groups management tool

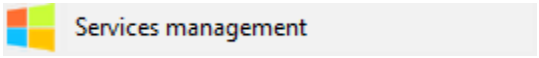


Fig. 25 – Menu option for remotely launching the Service Management tool on host

Lancer le téléchargement des mises à jours Windows en attente

The *Trigger the download of pending Windows Updates* menu option allows you to remotely trigger the download of pending Windows Updates on the selected host(s).

For more information, visit this *documentation*.

Lancer l'installation des mises à jours Windows en attente

The *Trigger the install of pending Windows Updates* menu option allows you to remotely trigger the installation of pending Windows Updates on the selected host(s).

For more information, visit this *documentation*.

Actualiser l'inventaire des postes

The *Refresh host inventory* menu option allows you to force selected host(s) to send their current inventory.

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
Launch update	Lancer l'action
Fermer	Ferme la fenêtre

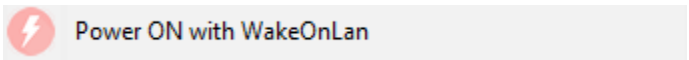


Fig. 26 – Menu option for remotely powering on hosts

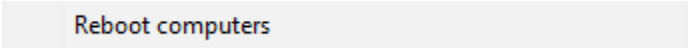


FIG. 27 – Menu option for remotely rebooting hosts

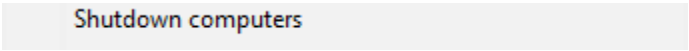


FIG. 28 – Menu option for remotely shutting down hosts

Lancer le redémarrage de waptservice

The *Trigger a restart of waptservice* menu option allows you to remotely force selected host(s) to restart their **`waptservice`**.

58.4 Sous onglet de l'hôte

58.4.1 Onglet Général

TABLEAU 1 – Informations displayed in the *Overview* tab

Name	Le nom de l'hôte
Description	La description de l'hôte
Operating system	Le système d'exploitation tournant sur l'hôte
IP address	L'adresse Ip de l'hôte
Last task	La dernière tâche WAPT qui s'est effectuée sur l'hôte
Manufacturer	Le constructeur de l'hôte
Model	Le modèle de l'hôte
Last seen	La date à laquelle la dernière mise à jour de l'hôte s'est passée
Logged in users	The name of the user last or currently connected on the host
UUID	L'UUID BIOS de l'hôte

58.5 Search keywords

58.6 Package status filtering

The checkboxes allow you to filter packages by their *status*.

FIG. 29 – Menu option for remotely triggering the scan for missing Windows Updates on hosts

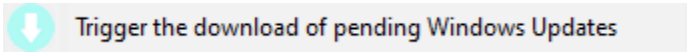


Fig. 30 – Menu option for remotely triggering the download of pending Windows Updates on hosts

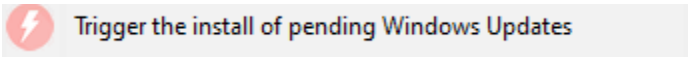


Fig. 31 – Menu option for remotely triggering the installation Windows Updates on hosts

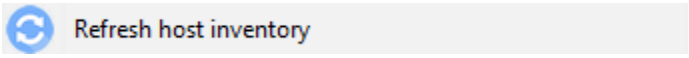


Fig. 32 – Menu option for refreshing the host inventory

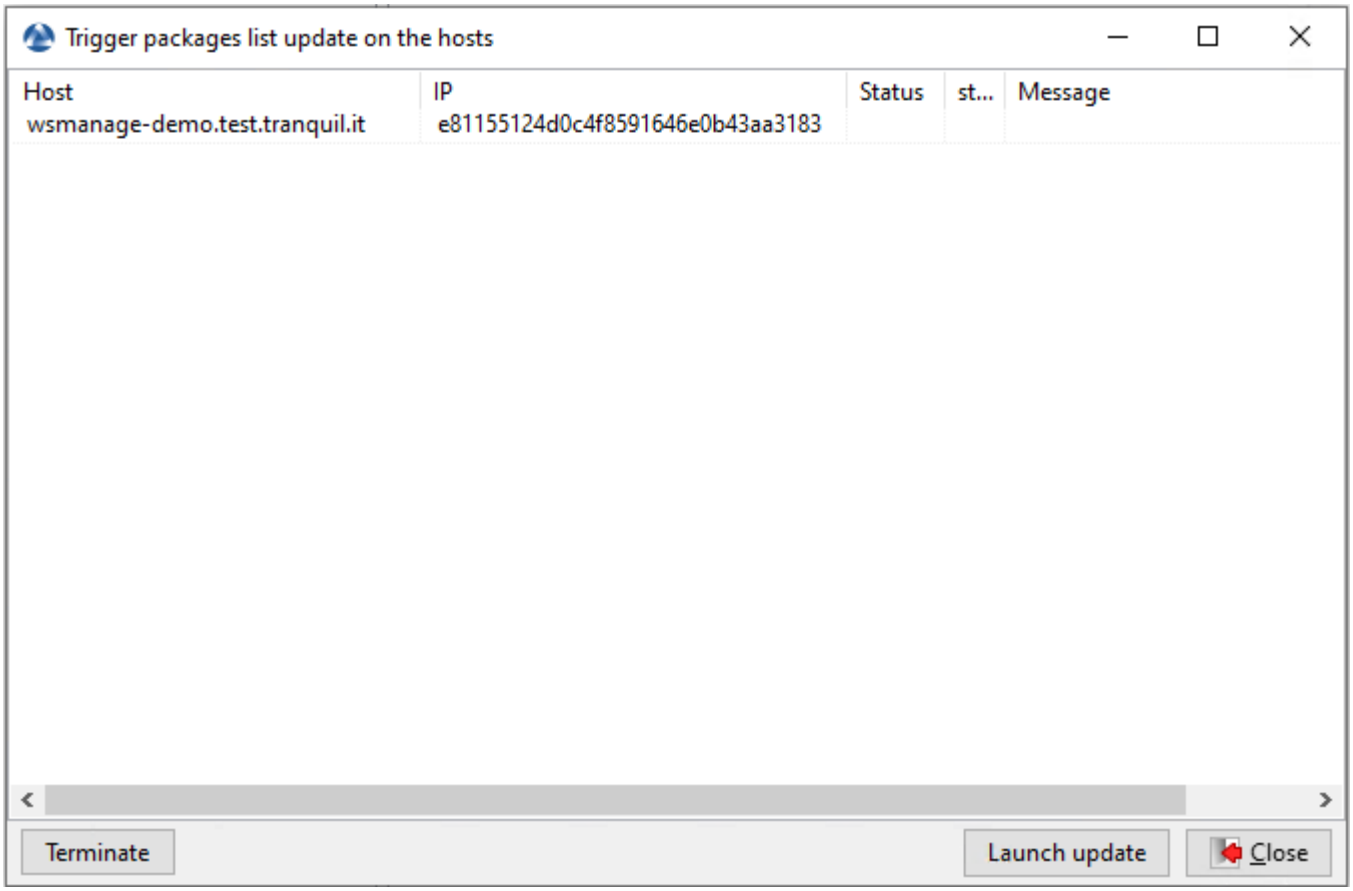


Fig. 33 – Window listing hosts currently updating their inventory

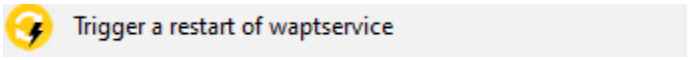
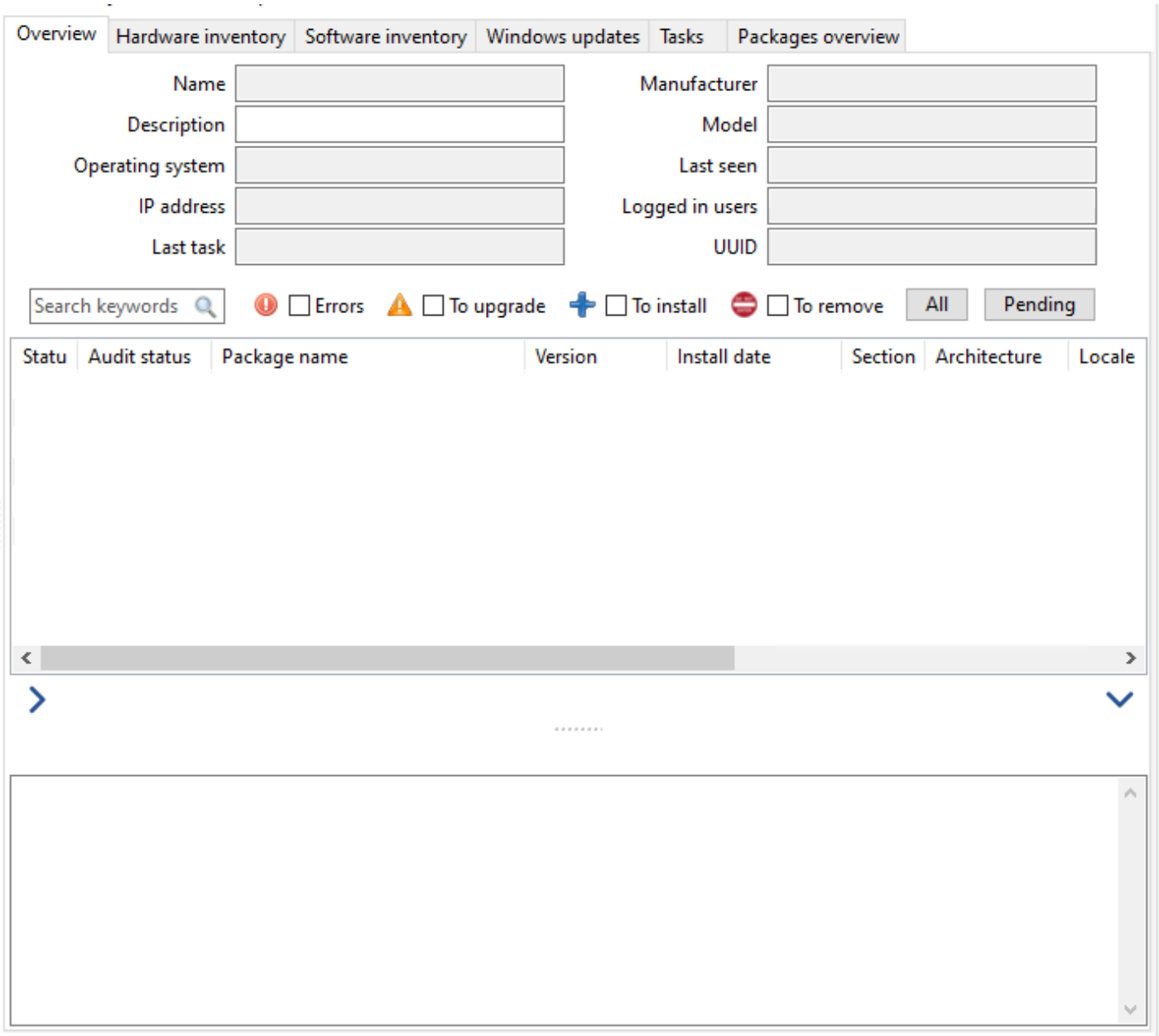
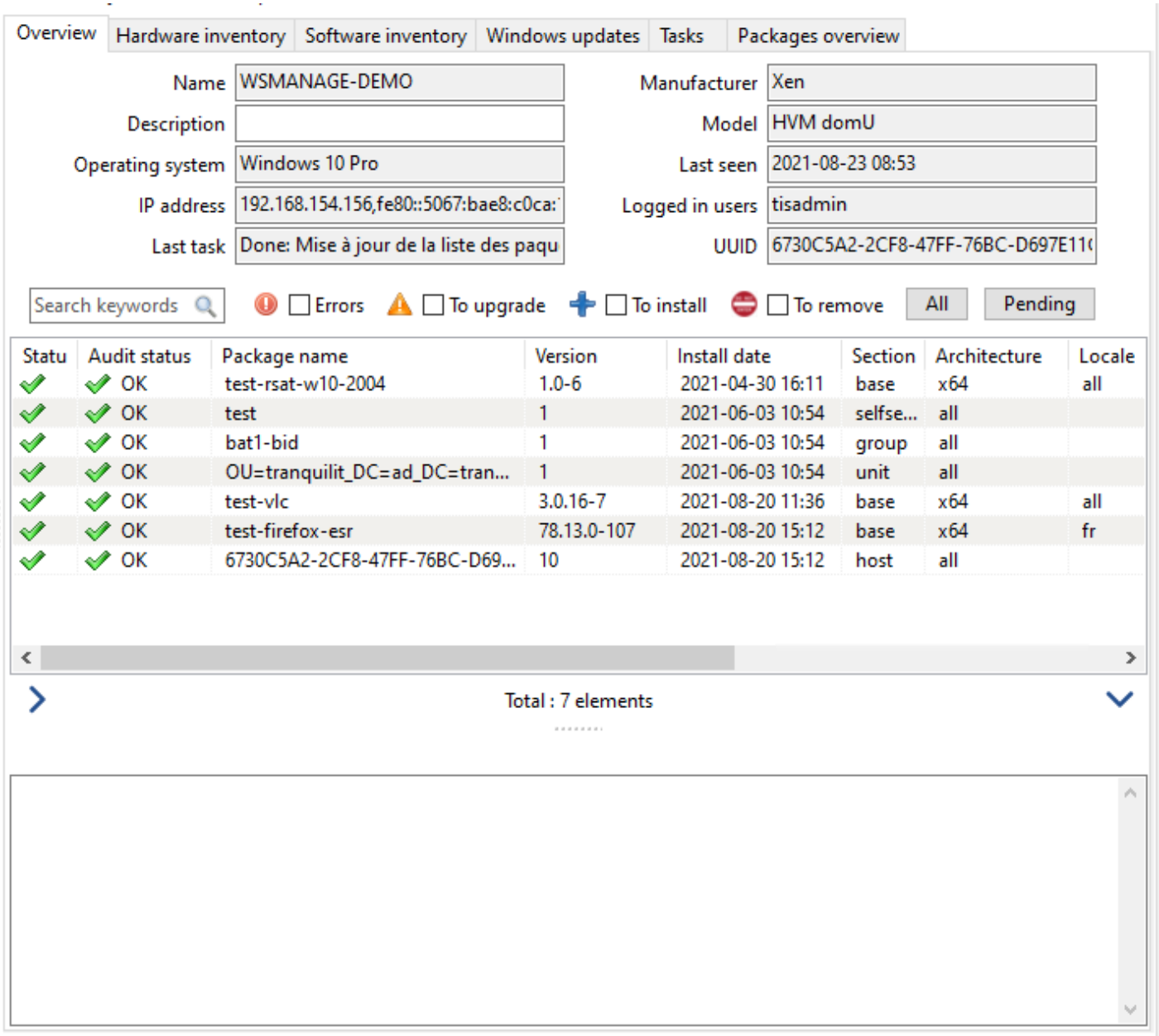
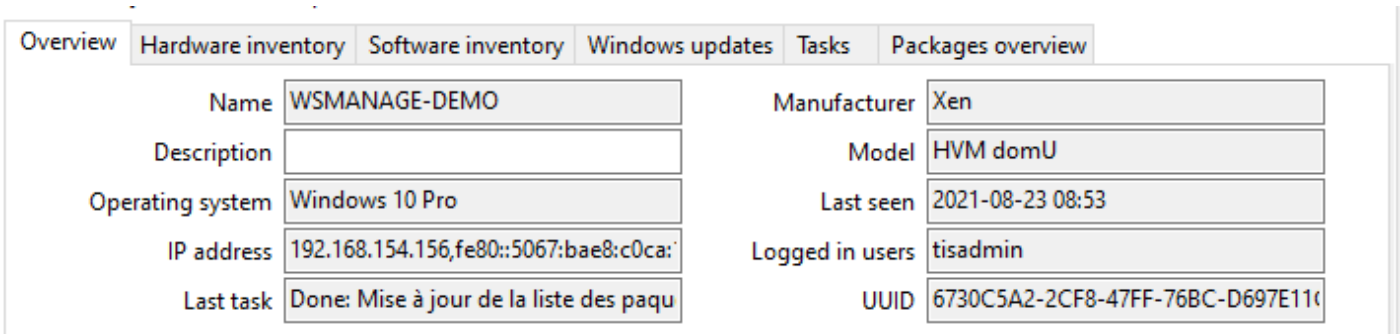


Fig. 34 – Menu option for remotely triggering a restart of the waptservice on hosts


Overview
Hardware inventory
Software inventory
Windows updates
Tasks
Packages overview
Name
Description
Operating system
IP address
Last task
Manufacturer
Model
Last seen
Logged in users
UUID
Search keywords
Errors
To upgrade
To install
To remove
All
Pending
Statu
Audit status
Package name
Version
Install date
Section
Architecture
Locale


Overview
Hardware inventory
Software inventory
Windows updates
Tasks
Packages overview
Name
WSMANAGE-DEMO
Description
Operating system
Windows 10 Pro
IP address
192.168.154.156,fe80::5067:bae8:c0ca:
Last task
Done: Mise à jour de la liste des paqu
Manufacturer
Xen
Model
HVM domU
Last seen
2021-08-23 08:53
Logged in users
tisadmin
UUID
6730C5A2-2CF8-47FF-76BC-D697E11(
Search keywords
Errors
To upgrade
To install
To remove
All
Pending
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


Overview
Hardware inventory
Software inventory
Windows updates
Tasks
Packages overview
Name
WSMANAGE-DEMO
Description
Operating system
Windows 10 Pro
IP address
192.168.154.156,fe80::5067:bae8:c0ca:
Last task
Done: Mise à jour de la liste des paqu
Manufacturer
Xen
Model
HVM domU
Last seen
2021-08-23 08:53
Logged in users
tisadmin
UUID
6730C5A2-2CF8-47FF-76BC-D697E11(

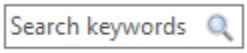


FIG. 35 – Field for searching on WAPT package keywords

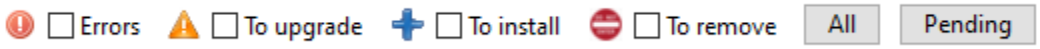


FIG. 36 – Checkboxes for filtering on host status

TABLEAU 2 – Status des paquets dans la console WAPT

Status	Description
Errors	List of packages that have not installed correctly.
To upgrade	List of packages for which an update is available.
To install	List of packages waiting to be installed.
To remove	List of packages waiting to be removed.

The checkbox *All* allows you to select all status.

The button *Pending* allows you to select packages having status :

— *NEED-UPGRADE* ;
— *NEED-INSTALL* ;
— *TO-REMOVE*.

58.7 Package layout

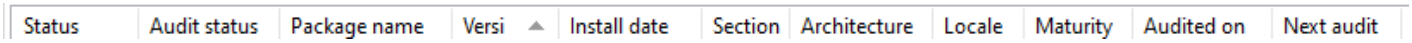


FIG. 37 – Header of the grid showing the list of WAPT packages installed on host

It is possible to select more columns with a Right-Click on the grid header.

TABLEAU 3: Status des paquets dans la console WAPT

Status	Description
Status (install_status)	*Status* of the WAPT package.
Audit status (last_audit_status)	*Last date of compliance audit*.
Version (version)	Version of the WAPT package.
Description (description)	Description of the WAPT package.
Dependencies (depends)	Dependancies of the WAPT package.
Install date (install_date)	Install date of the WAPT package.
priority	**Not used**.
sources (sources)	URL for downloading the original software load.
Installed by (explicit_by)	User who has installed the package on the host.
Section (section)	*Type* of WAPT package.
Architecture (architecture)	CPU architecture of the package.
Locale (locale)	Language of the WAPT package.

suite sur la page suivante

Tableau 3 – suite de la page précédente

Status	Description
md5sum (md5sum)	md5 sum of the WAPT package.
File name (filename)	Name of the package on the WAPT repository.
Parameters (install_params)	List of installation parameters for the WAPT package.
Log install (install_output)	Installation logs of the WAPT package.
maintainer (maintainer)	Name of the maintainer of the WAPT package.
conflict (conflict)	List of forbidden packages (packages that need to be uninstalled before the package can install).
Size after install (installed_size)	Storage occupied by the software once installed.
URL of repository (repo_url)	URL of the repository from which the package has been downloaded.
Size (size)	Size of the WAPT package in the repository.
Name of repository (repo_url)	FQDN of the repository from which the package has been downloaded.
Maturity (maturity)	Level of maturity of the WWAPT package (**DEV**, **PREPROD**, **PROD**).
Audited on	Date of the last compliance audit.
Next audit (next_audit_on)	Date of the next compliance audit.
uninstall_key (uninstall_key)	Uninstall key of the software.
update_on (update_on)	Date of the last package update.
update_by (update_by)	User having last updated the package.
created_on (created_on)	Date of first installation of the WAPT package.
created_by (created_by)	User having first installed the WAPT package.
last_audit_output (last_audit_output)	Log of the last compliance audit
id (id)	Database ID of the package in the WAPT agent local `sqlite`.

58.8 Package status

Tableau 4 – Status des paquets dans la console WAPT

Status	Description
OK	Paquet installé
ERROR	Paquet ayant subit une erreur d'installation
NEED-UPGRADE	Paquet ayant une mise à jour en attente
NEED-INSTALL	Paquet en attente d'installation
TO-REMOVE	Package with pending uninstall

It is possible to get informations about status by clicking on it.

For example, here is an error status :

58.9 Acting on packages installed on a host

Indication :
- Selecting several packages is possible ;
- If several hosts are selected, the action will be launched on all selected hosts ;

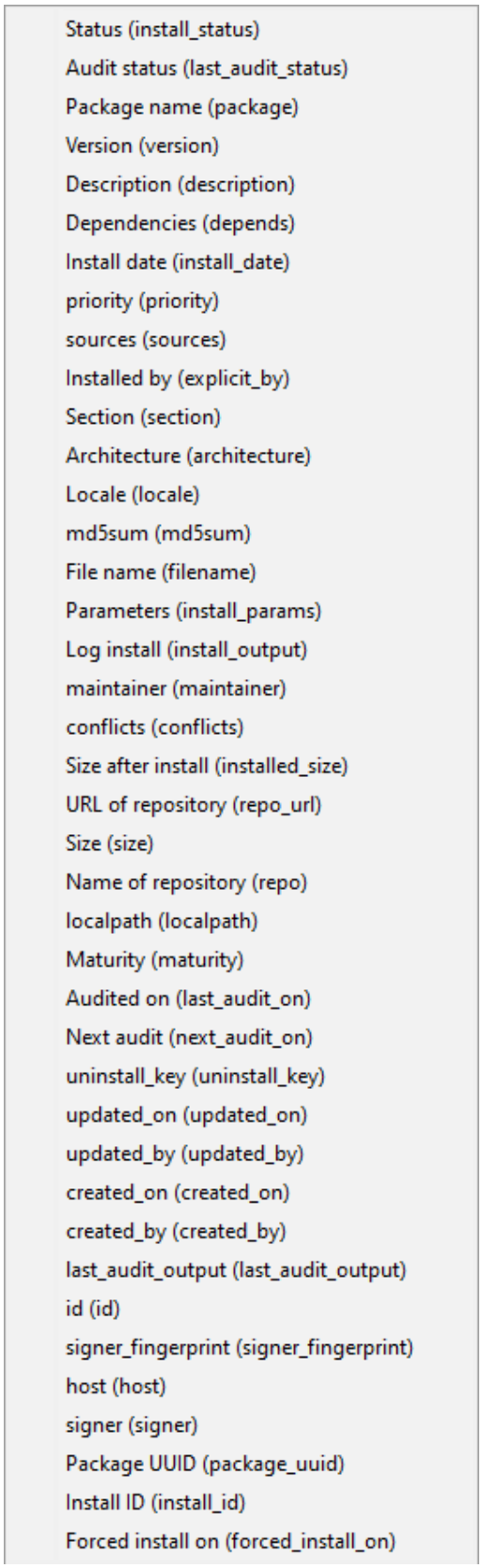
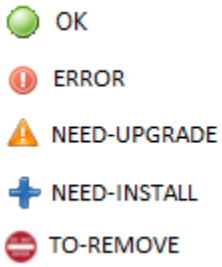
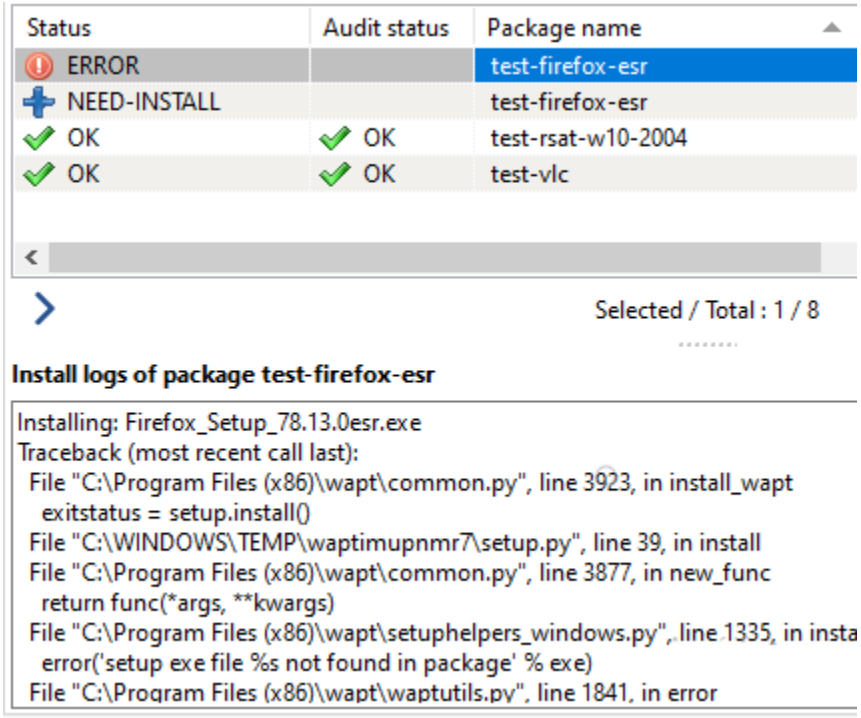


FIG. 38 – List of available columns to add to the grid


OK
ERROR
NEED-UPGRADE
NEED-INSTALL
TO-REMOVE


Status
Audit status
Package name
ERROR
test-firefox-esr
NEED-INSTALL
test-firefox-esr
OK
OK
test-rsat-w10-2004
OK
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\setuphelpers_windows.py", line 1335, in insta
error('setup exe file %s not found in package' % exe)
File "C:\Program Files (x86)\wapt\waptutils.py", line 1841, in error

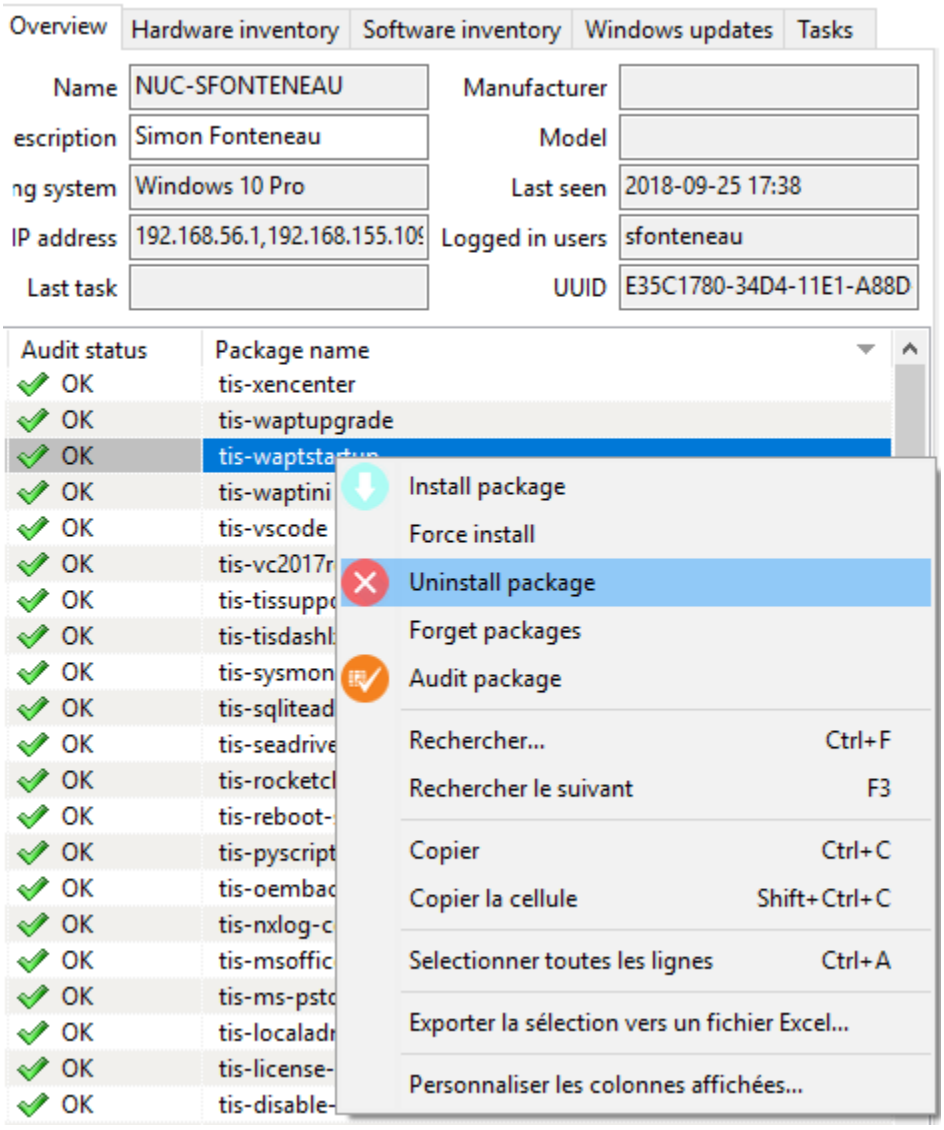


FIG. 39 – Possible actions for WAPT packages

Tableau 5 – Available parameters for the [option] section of waptserver.ini

Action	Description
Install a package	The action installs the selected package on the selected host(s).
Force a package	The action forces the re-installation of a selected package on the selected host(s).
Remove a package	The action removes the selected package from the selected host(s)
Forget a package	The action tells the selected host(s) **to no longer use WAPT** to manage the software or the configuration
Audit the package	The action triggers an audit on the selected package(s)

58.10 Onglet Inventaire Matériel

Les informations affichées par défaut dans l'*Inventaire Matériel* sont :

- Information on the host's hardware components.
- Information about the host.
- Information on the status of 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 aparraître dans la grille principale.

58.11 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*.

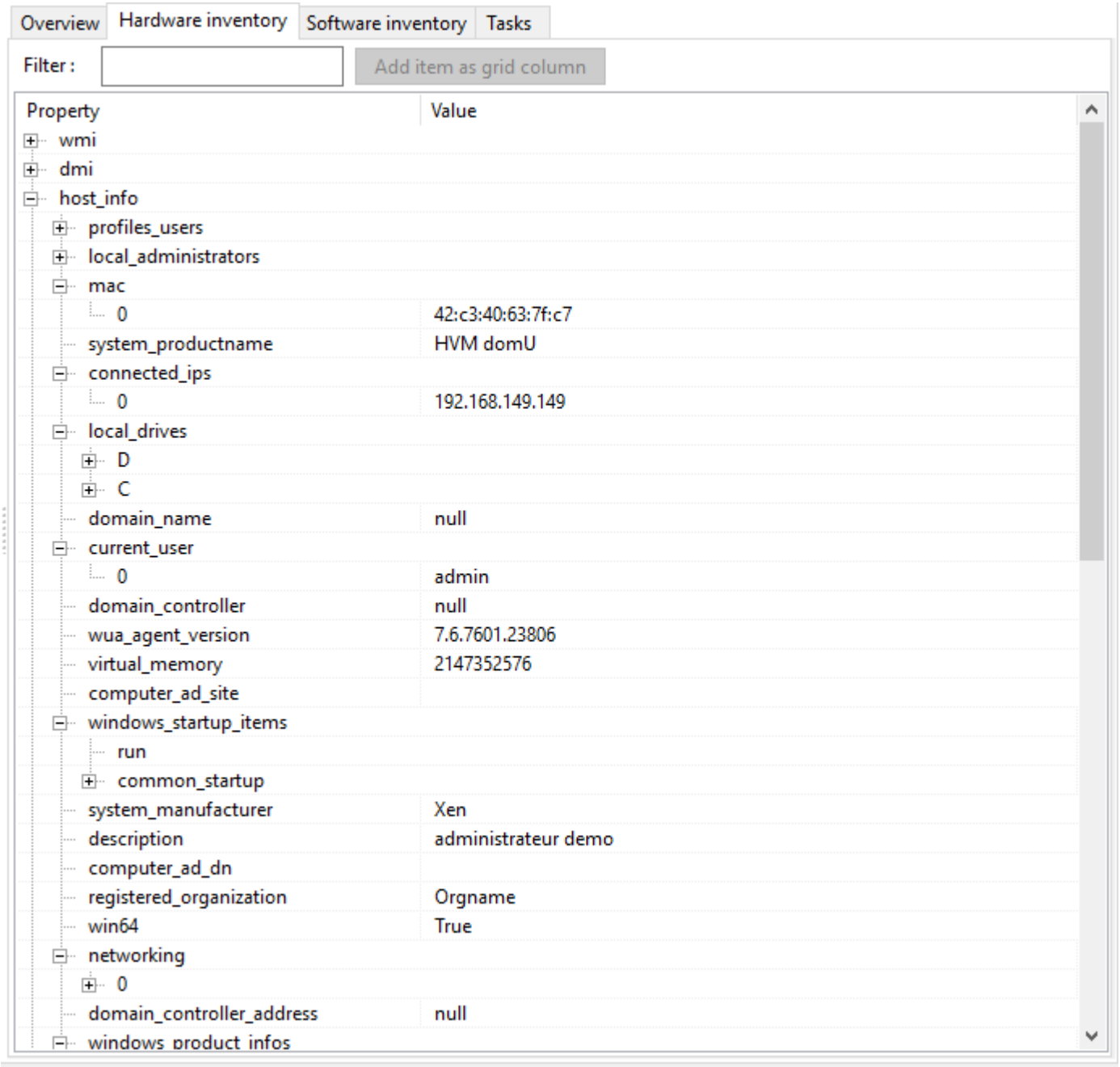


FIG. 40 – L'inventaire Matériel de l'hôte

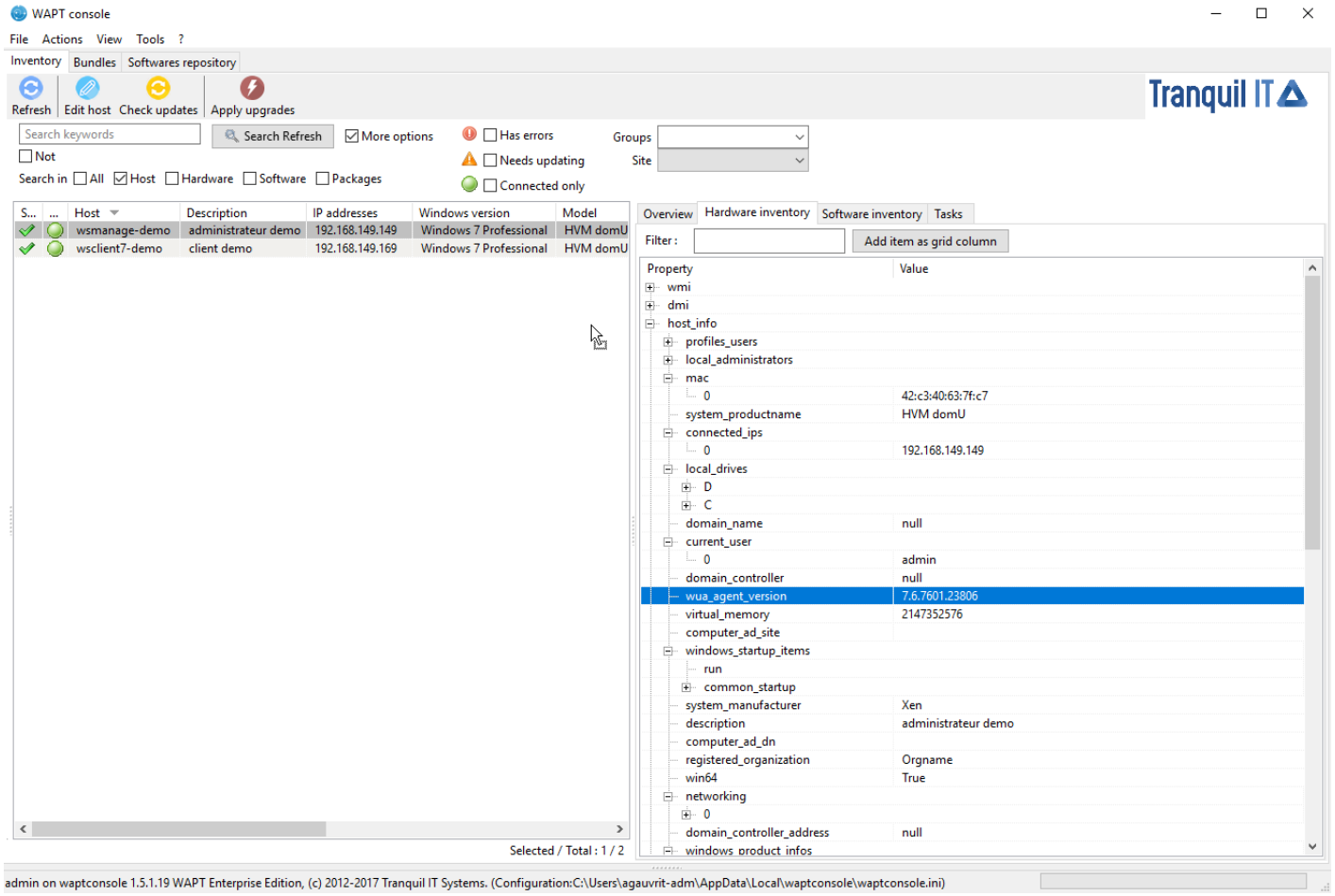


FIG. 41 – Ajouter un critère à la grille principale de la console WAPT

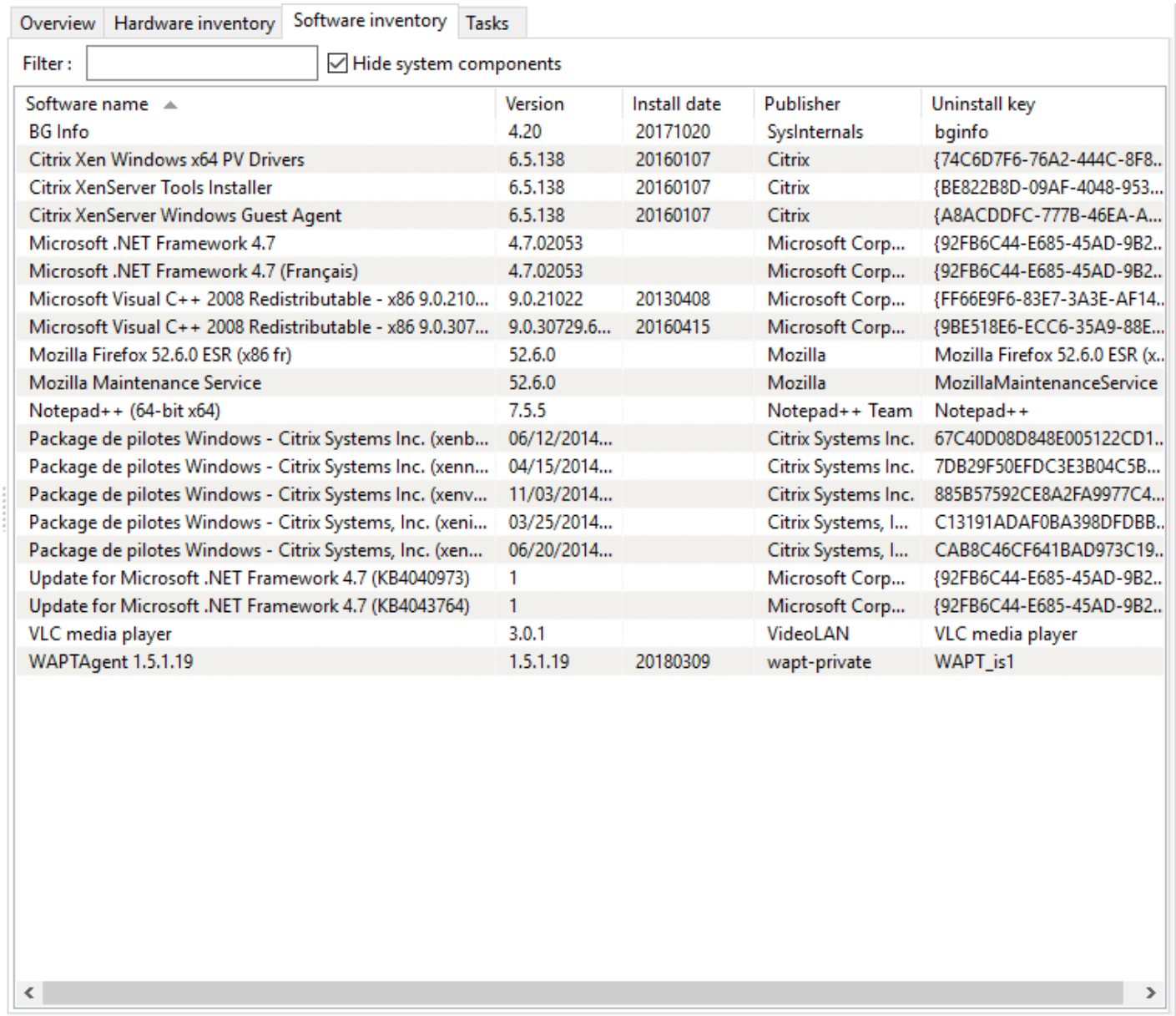


FIG. 42 – L'inventaire logiciel est contenu dans la base de registre Windows de l'hôte

58.12 L'onglet Windows update

Les informations affichées dans l'onglet *Windows update* sont :

- Windows update agent version.
- Date of the last Windows update scan.
- Duration of the last scan.
- WAPTWUA status.
- Date of the last version of `wsusscn2.cab` processed by WAPT.
- Status of WAPTWUA Enabled (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* ;
- *Criticté* ;
- *Classification* ;
- *Produit* ;
- *Download size*.

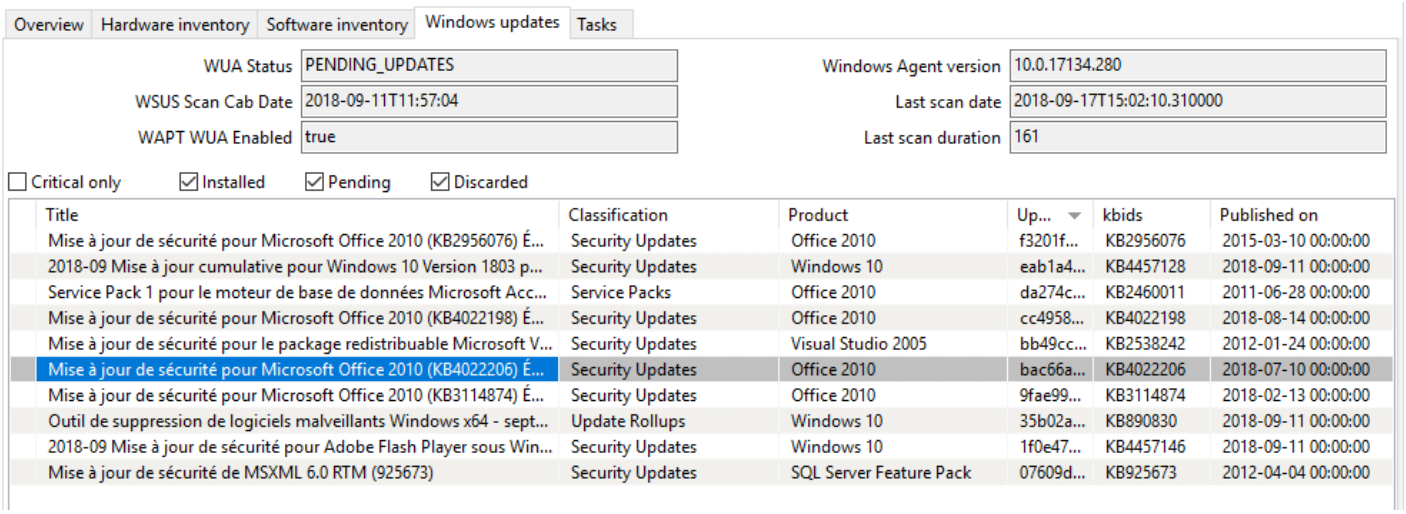


Fig. 43 – Inventaire des Windows Updates

58.13 L'onglet tâche

Les informations affichées par défaut dans l'onglet *Tâches* sont :

- Pending tasks.
- Completed tasks.
- Tasks in error.

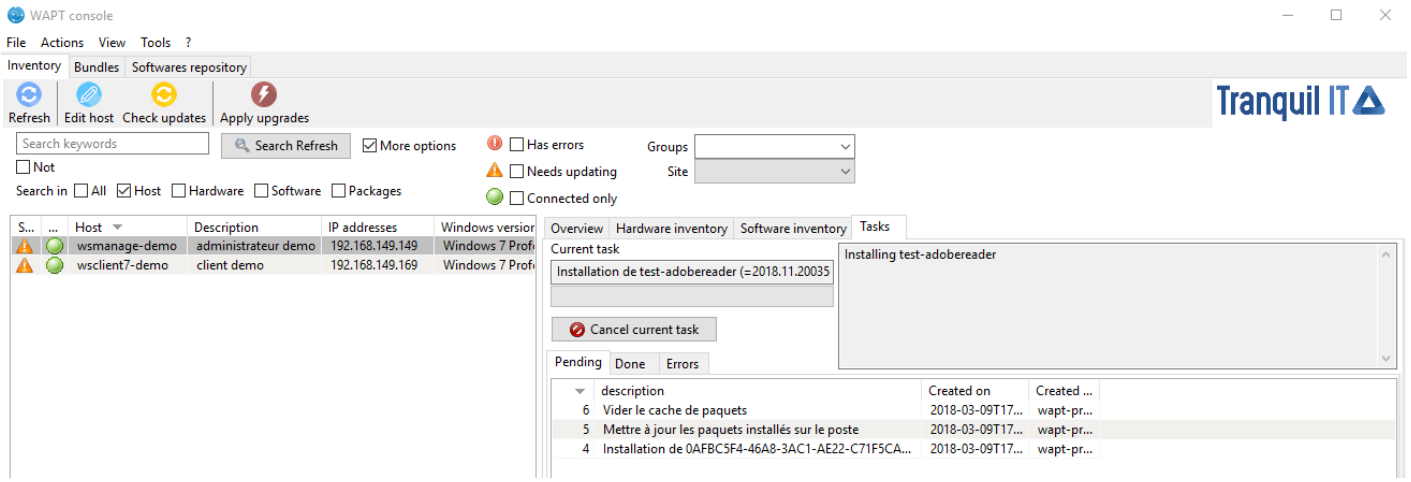


FIG. 44 – Détails des tâches en attente sur l'hôte

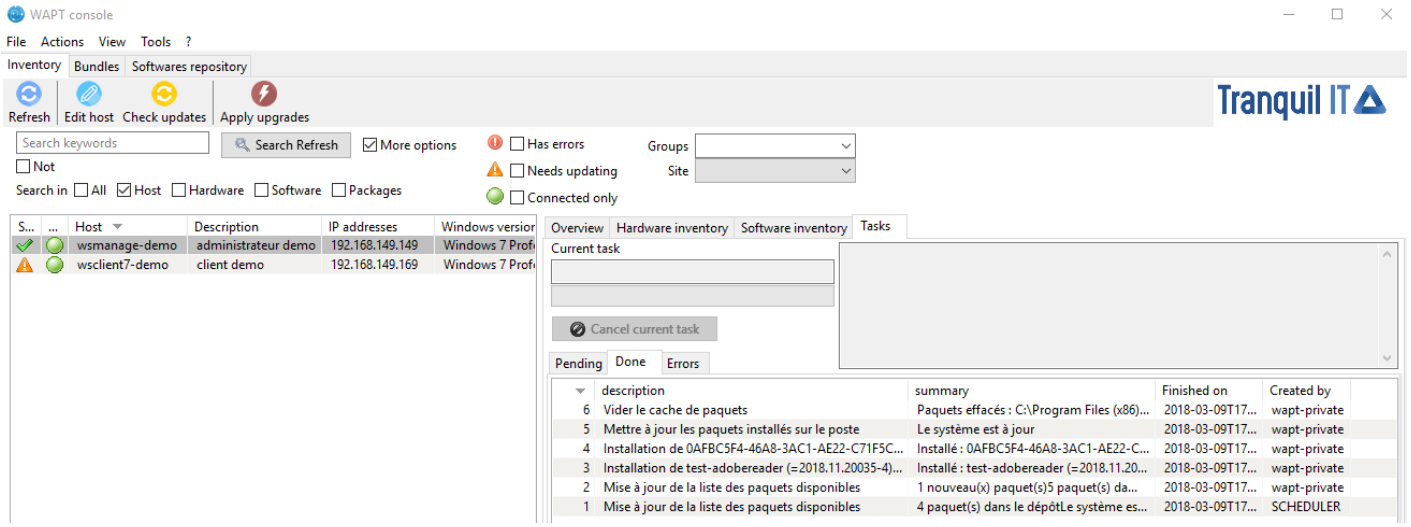


FIG. 45 – Détails des tâches complétées

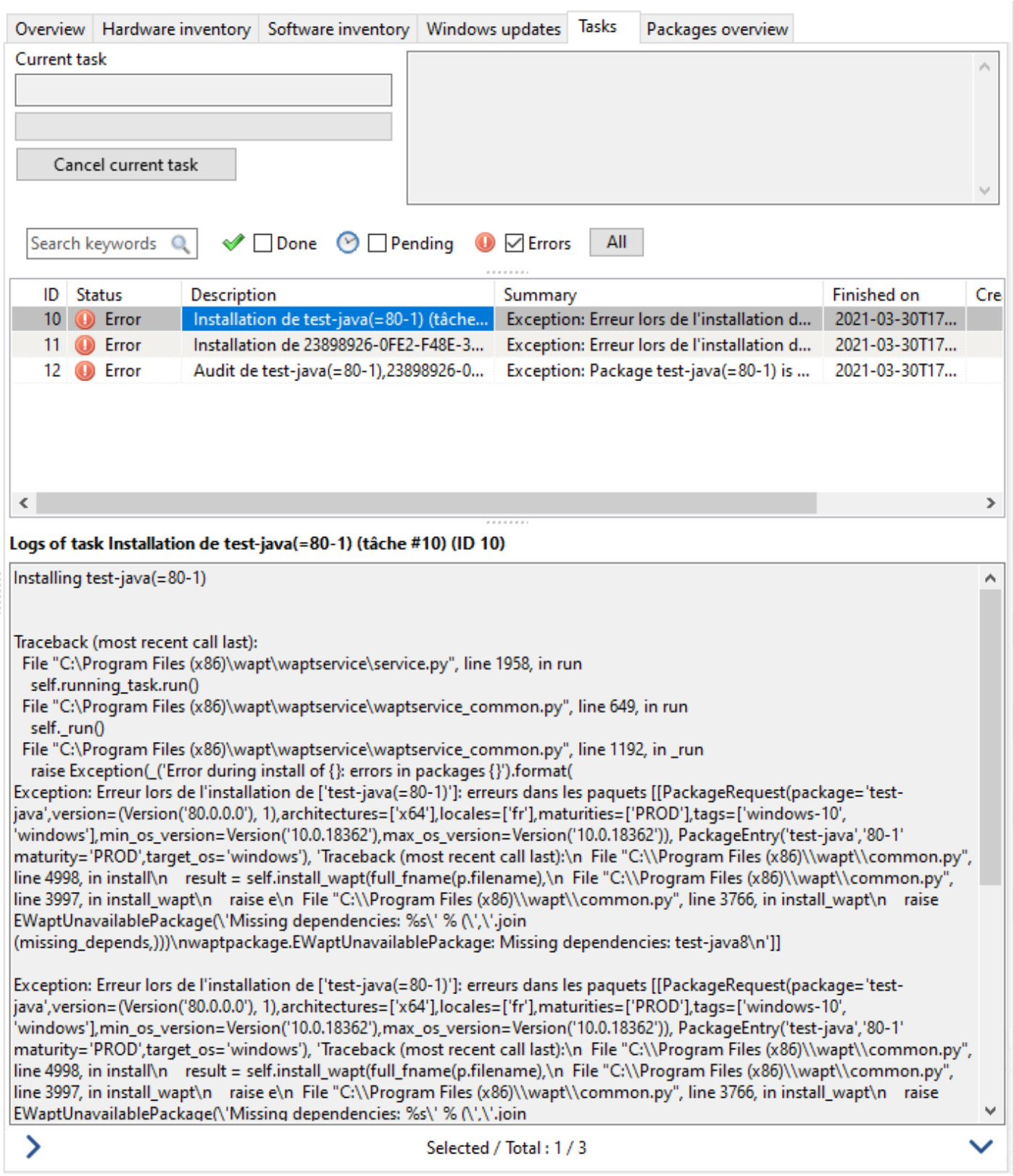


Fig. 46 – Détails des tâches en erreur

58.14 Vue global des paquets

58.15 Données d'audit

CHAPITRE 59

L'onglet Paquets WAPT

59.1 Rafraîchir la liste des paquets

59.2 Importer des paquets

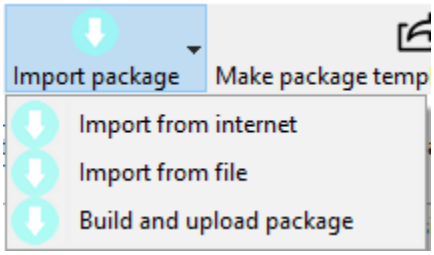


FIG. 1 – Button for importing a WAPT package

Attention : If the verification of *the package signature* is enabled for added security, the public certificate of the signer(s) whose packages are stored in the repository must be located in one of the following folders :

— `C:\Program Files (x86)\wapt\ssl`;
— `%appdata%\waptconsole\ssl`;

If the certificate is not found in one of these two folders, then the following error will occur and no package will not be imported from the repository.

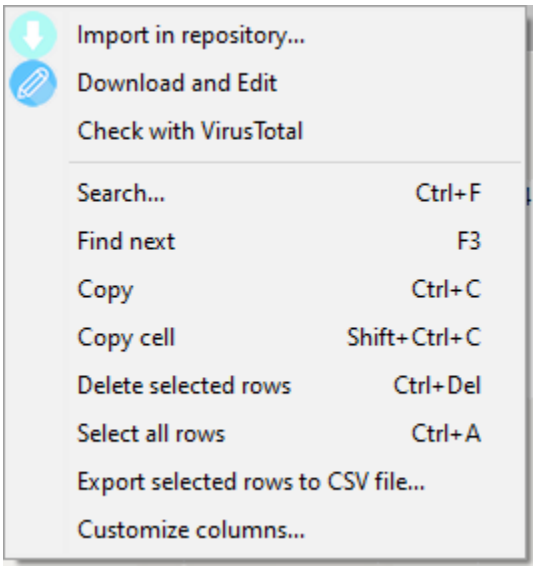


FIG. 2 – Menu for importing a WAPT package

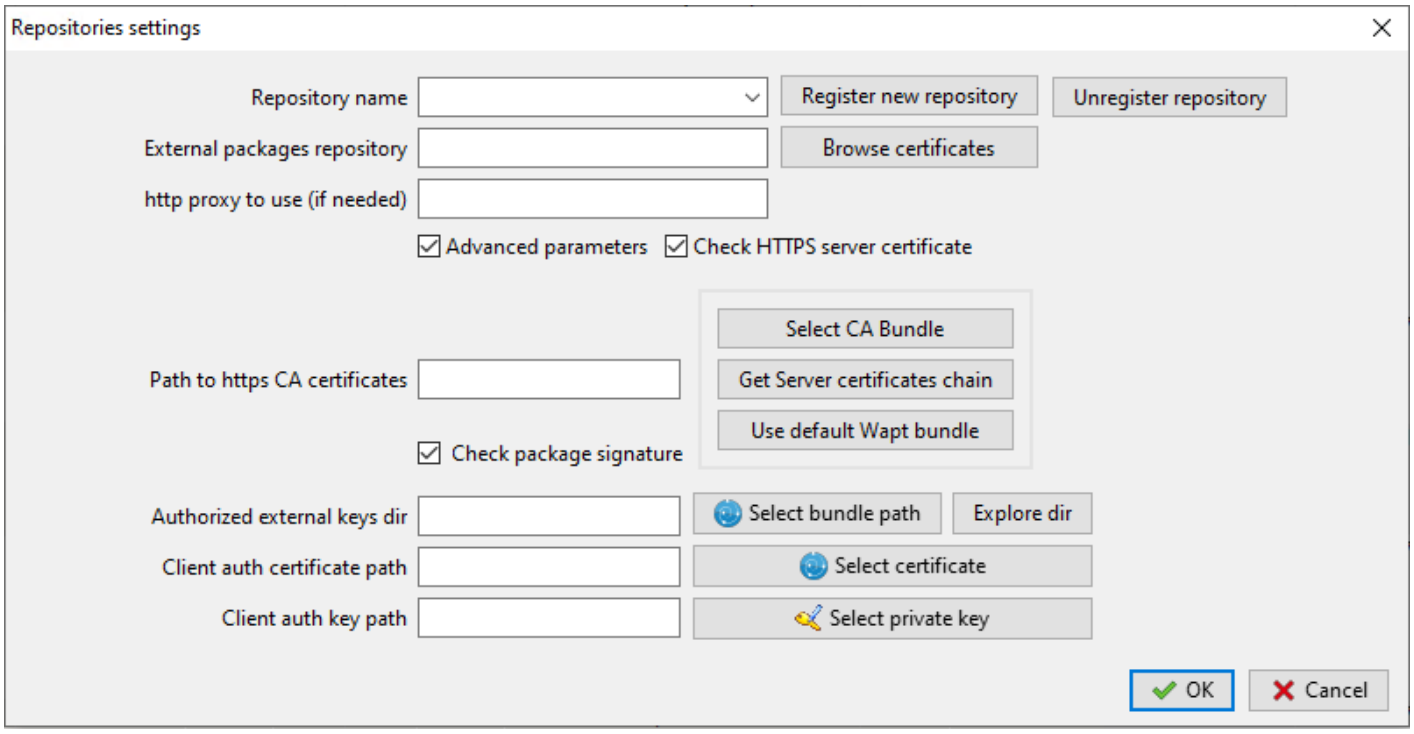


FIG. 3 – Window for setting up a repository to import packages from

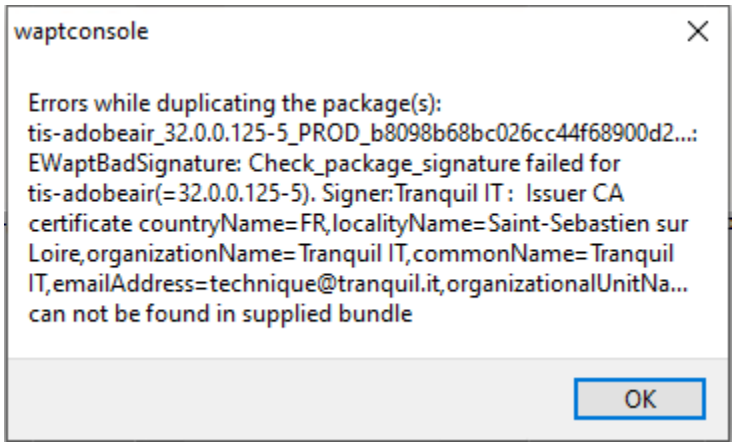
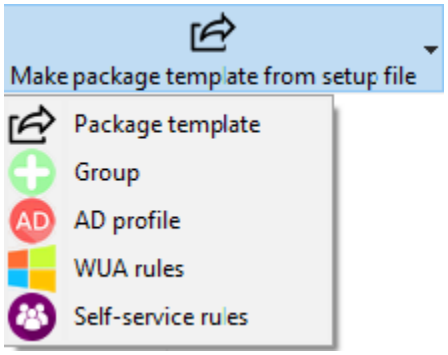


FIG. 4 – Error while validating the signature of an external repository

59.3 Import from file

59.3.1 Construire et téléverser les paquets

59.4 Créer des modèles de paquets dà partir d'un fichier setup



59.4.1 Modèle de paquet

To select file, click on :

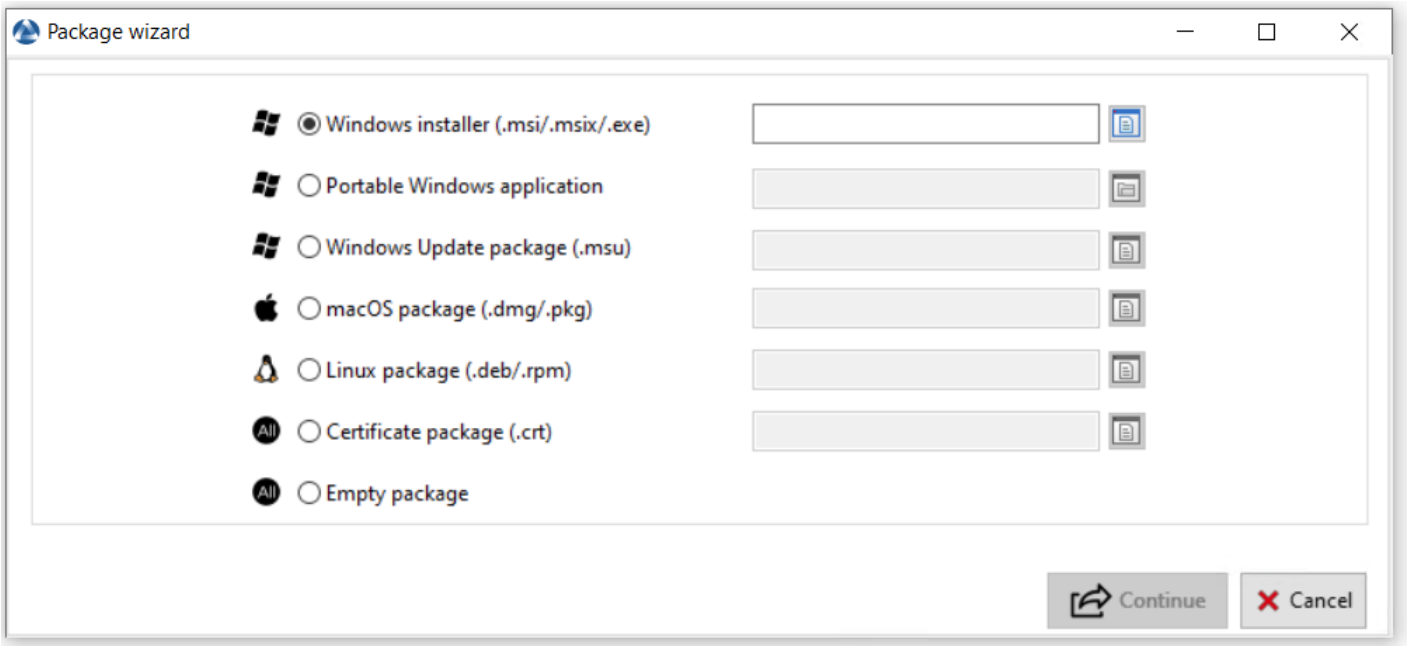


FIG. 5 – Assistant création de paquet

Installeur Windows (.msi/.msix/.exe)

Application portable Windows

Paquet Windows Update (.msu)

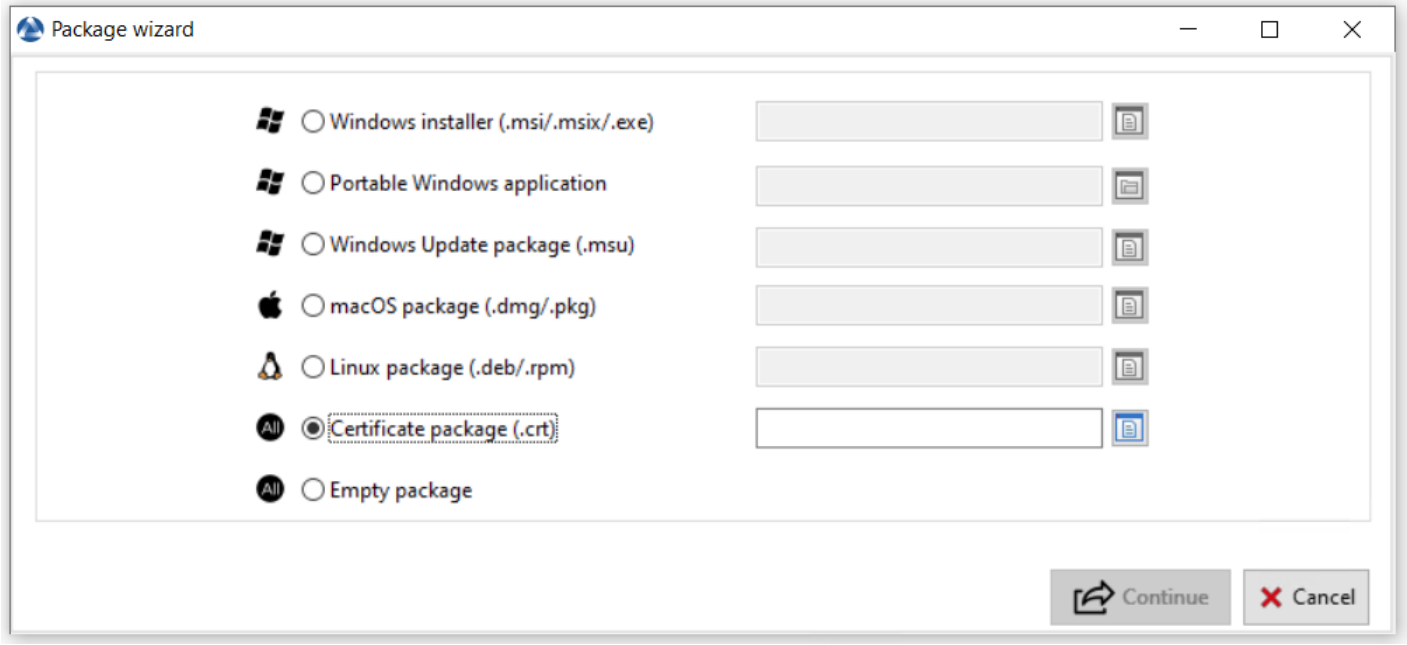
paquet macOS (.dmg/.pkg)

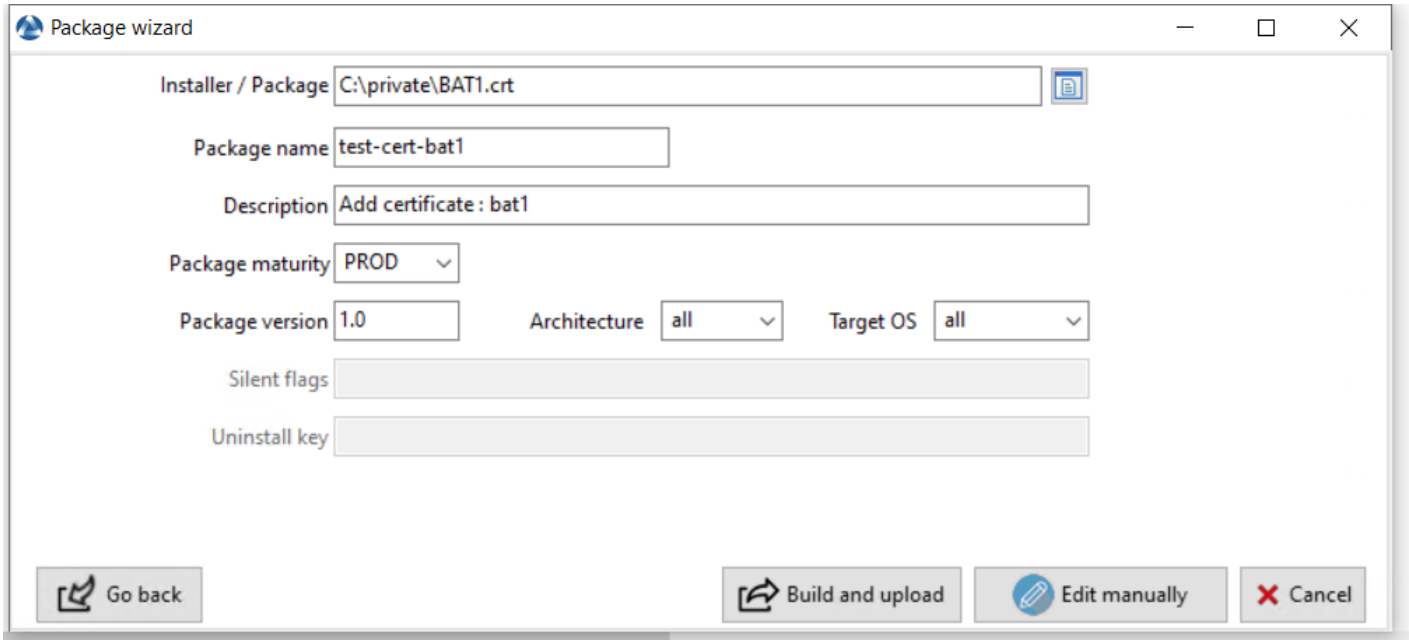
Paquet Linux (.deb/rpm)

Paquet de certificate (.crt)

Sélectionnez .crt puis cliquez sur `Continuer`

Conformer les informations puis cliquez sur `Créer et téléverser`

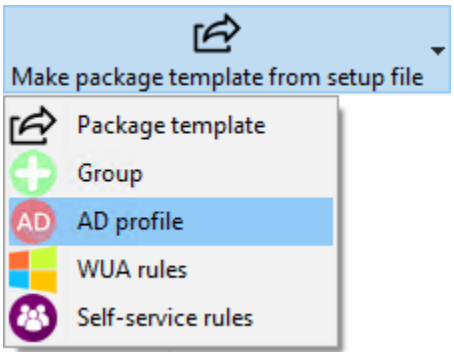

Package wizard
Windows installer (.msi/.msix/.exe)
Portable Windows application
Windows Update package (.msu)
macOS package (.dmg/.pkg)
Linux package (.deb/.rpm)
Certificate package (.crt)
Empty package
Continue
Cancel


Package wizard
Installer / Package
C:\private\BAT1.crt
Package name
test-cert-bat1
Description
Add certificate : bat1
Package maturity
PROD
Package version
1.0
Architecture
all
Target OS
all
Silent flags
Uninstall key
Go back
Build and upload
Edit manually
Cancel

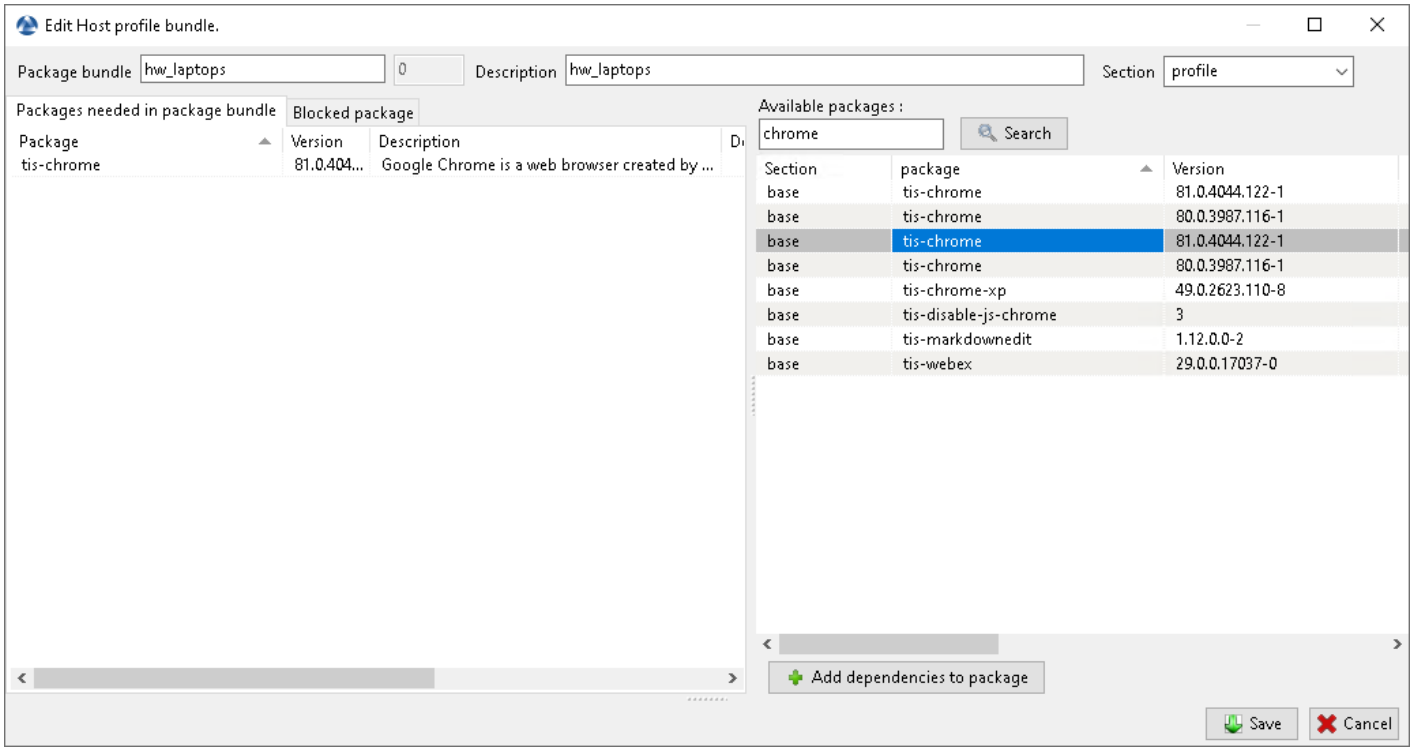
Paquet vide

59.4.2 Groupe

59.4.3 Profile AD



After clicking :



For using, see *AD profile <wapt_profile_bundles>*

59.4.4 Règles WUA

59.4.5 Règles Self-service

59.5 Search tab

59.6 Dernière version uniquement

59.7 Filtres des paquets

59.8 Montrer les hôtes

59.9 Architecture

59.10 OS

59.11 Locale

59.12 Maturité

59.13 Action sur les paquets

59.13.1 Surpprimer du dépôt

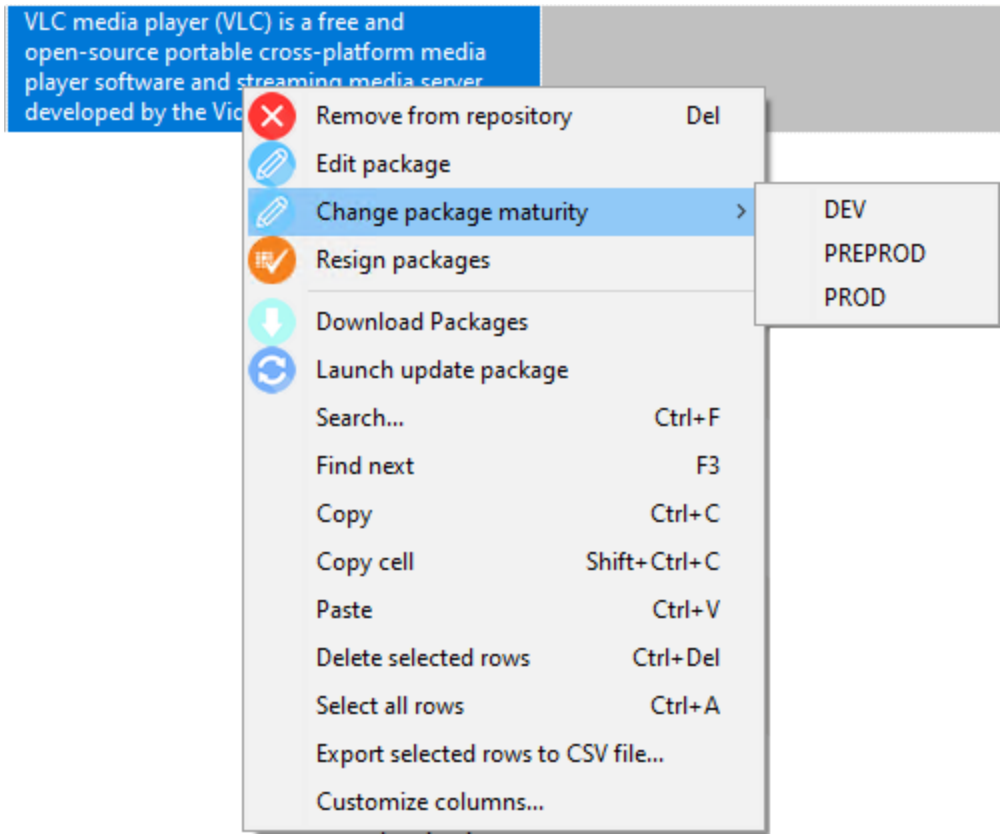
59.13.2 Editer des paquets

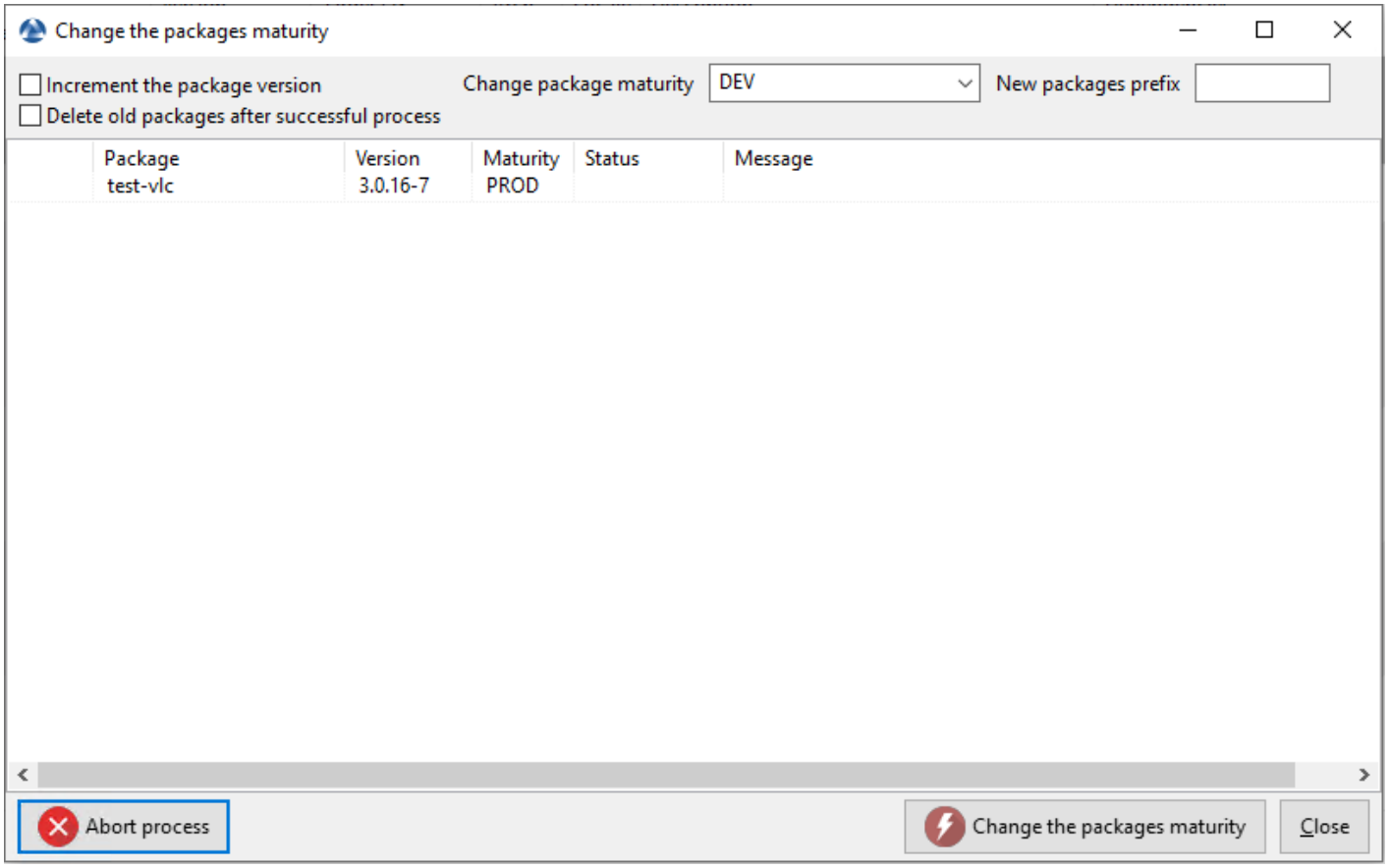
59.13.3 Changer la maturité d'un paque

59.13.4 Resigner les paquets

59.13.5 Télécharger les paquets

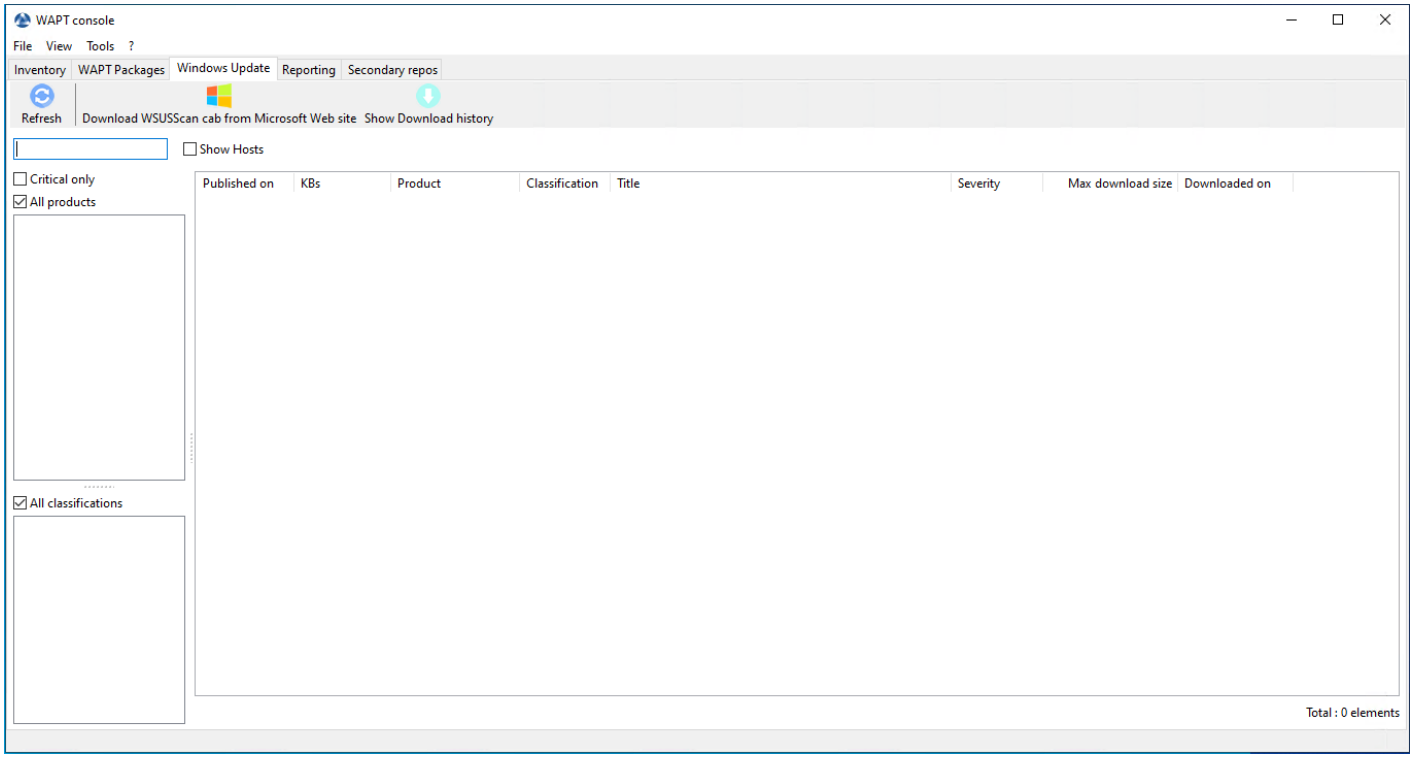
59.13.6 Rechercher


VLC media player (VLC) is a free and
open-source portable cross-platform media
player software and streaming media server
developed by the Vic
Remove from repository Del
Edit package
Change package maturity
Resign packages
Download Packages
Launch update package
Search... Ctrl+F
Find next F3
Copy Ctrl+C
Copy cell Shift+Ctrl+C
Paste Ctrl+V
Delete selected rows Ctrl+Del
Select all rows Ctrl+A
Export selected rows to CSV file...
Customize columns...
DEV
PREPROD
PROD


Change the packages maturity
Increment the package version
Delete old packages after successful process
Change package maturity
DEV
New packages prefix
Package
Version
Maturity
Status
Message
test-vlc
3.0.16-7
PROD
Abort process
Change the packages maturity
Close

CHAPITRE 60

Onglet Windows Update



60.1 Onglet Source

60.2 Onglet Reporting

60.2.1 Rafraîchir

60.2.2 Design mode

60.2.3 Exéciter

60.2.4 Export to spreadsheet

60.2.5 Normalize software titles

60.2.6 Right click menu

Exéciter

Edit name

Export to spreadsheet

Export queries to file

Import queries

Rechercher

CHAPITRE 61

Secondary repos tab

61.1 Left on tab

61.1.1 Rafraîchir

61.1.2 Up rule

61.1.3 Down rule

61.1.4 Add rule

61.1.5 Edit rule

61.1.6 Delete rule

61.1.7 Save rule changes

61.1.8 Import Rules

61.1.9 Export Rules

61.1.10 Right click menu

Edit rule

Delete rule

61.2 Right on tab

61.2.1 Rafraîchir

61.2.2 Sync selected

61.2.3 Sync all

61.2.4 Create index

61.2.5 Changelog

61.2.6 Search input

61.2.7 Right click menu

Sync selector

Force sync selected

Show errors

Check files

Vous trouverez dans cette section de la documentation quelques méthodes inspirantes qui vous feront gagner du temps en vous permettant de tirer le meilleur de votre installation WAPT.

CHAPITRE 62

Simplifier le clônage de vos postes de travail

On constate que de nombreuses entreprises et administrations intègrent des logiciels et des configurations dans les images Windows qu'elles déploient sur leur parc de machines.

Si vous utilisez WAPT, perdez cette habitude maintenant et pour toujours ! Pourquoi ?

— 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.
— Each time you make a new image, if you are serious about it, you will have to keep track of the changes in a text document, a spreadsheet, or a change management tool. It's a very heavy and thankless burden. And you know as well as I do, what's ungrateful is usually badly done !
— Finally, if you introduce in your image security configurations, network configurations, or configurations to limit the intrusion of Windows telemetry, these configurations can disrupt the normal functioning of WAPT, it will complicate future diagnostics, and it will discourage you from using an efficacious tool very capable of freeing up your time.

62.1 Que proposez-vous de faire alors ?

Tranquil IT recommande :

— To make only one raw image per OS type with MDT or Fog (win10, win2016, etc) without any configuration or software. **Put only the system drivers** you need for your image deployment in the MDT or Fog directories provided for this purpose ;
— To configure your WAPT server to register hosts with a random UUID to avoid *UUID Bios or FQDN conflicts* ;
— To create as many Organizational Units as you have machine types in the *CN=Computers* OU (ex : *standard_laptop*, *hardened_laptop*, *workstations*, *servers*, etc) in your Active Directory ;
— To configure your Active Directory to distribute the WAPT Agent by GPO to the different Host Organizational Units ; This way, you can opt for fine grained configurations of your `waptagent.ini` for the hosts attached to each OU.

Note : Vous pouvez également inclure un agent WAPT générique dans votre image de système d'exploitation.

— To properly configure your DHCP to redirect the PXE to the correct system images ;
— To properly configure your MDT or Fog to register the machine in the correct Organizational Unit of your Active Directory ;

— To create as many WAPT security configuration packages as you have Organizational Units created above. Thus, you will be able to apply different security profiles depending on the type of machine. These packages will include the desired security configurations (telemetry suppression, firewall configuration, etc) ;

Indication : To save you time, you can base your security configuration strategy on security WAPT packages already available in the WAPT Store, you will only need to complete them according to your Organization's specific security requirements.

— To create in the *CN=Computers* OU as many Organizational Units as there are types of computer usage in your organization (*accounting*, *point_of_sale*, *engineering*, *sedentary_sales*, etc).
— To create generic WAPT packages of your software applications with their associated configurations.

Note : Pour vous faire gagner du temps et des efforts, vous pouvez importer de nombreux paquets WAPT éprouvés des magasins publics de Tranquil IT ou vous abonner aux magasins privés de Tranquil IT.

You can save even more time and effort if you make a judicious use of OU to model your fleet of computers according on their purpose.

62.1.1 Comment le scénario fonctionne-t-il ?

— You receive or the IT manager at the remote site receives a new machine in its box.

Indication : Vous pouvez également choisir ou le responsable informatique du site distant choisit de faire passer une machine existante de win7 à win10. Vous aurez, ou il aura préalablement sauvegardé le(s) répertoire(s) de l'utilisateur sur un lecteur réseau ou un autre support de stockage pratique.

For this purpose, you may build a WAPT package that, upon execution, will zip the `C:Users` on the win7 computer, name it with the computer's FQDN, password protect the compressed file using *this procedure* and upload it to a web server or a network share. That same WAPT package can do the reverse process and reinstall the user files after the host has been re-imaged.

— You configure MDT or Fog with the machine's MAC address so that it gets the right system image through DHCP and is positioned in the right Organizational Unit at the end of the cloning process.
— The expected system image is downloaded on the machine in masked time, the machine is placed in the right Organizational Unit.
— The WAPT agent registers the machine with the WAPT server, it appears in the WAPT console.

Indication : If your machines are from a win7 to win10 update, then you will remove the old win7 machines from the WAPT inventory as they will be duplicated due to your choice of random UUID configuration ; these machines will be easy to find in the WAPT console because they will be marked as win7 with the same MAC address or the same FQDN as your new machine in win10 ; after removing the win7, your inventory will be clean and up to date in your WAPT console.

— The WAPT agent detects that it is in an Organizational Unit that requires a particular software set and a particular security configuration.
— The WAPT Agent downloads and executes software packages and security configuration packages in hidden time ; the WAPT Agent automatically removes delegated rights that are rendered useless after joining the domain to prevent them from being subsequently exploited in an unauthorized manner.
— Either by group of machines or machine by machine, you finalize the configuration of the machines by assigning specific WAPT packets to them.

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).

62.1.2 Conclusion

Avec peu d'efforts, vous avez maintenant le contrôle total d'une flotte de plusieurs centaines, voire milliers de machines dispersées géographiquement. Toutes vos installations sont documentées, vos utilisateurs travaillent avec des droits adéquats et vous bénéficiez d'une visibilité claire sur les outils et les usages de vos utilisateurs. Ainsi, le passé n'est plus un fardeau impondérable pour vous et un obstacle à vos projets futurs.

CHAPITRE 63

Problèmes fréquents

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.

63.1 Réinitialiser le mot de passe du serveur WAPT Linux

- se connecter au serveur avec SSH ;
- se connecter avec l'utilisateur root (ou utiliser sudo) ;
- lancer le script de post-configuration :

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

Attention : Pour éviter de casser la configuration existante du serveur WAPT, acceptez toutes les autres étapes, **NE CRÉEZ PAS une nouvelle clé privée** !

63.2 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 regénérer une nouvelle clé et les certifcats 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.

63.2.1 Procédure de renouvellement ou de création d'une clé privée

La procédure va être la suivante :

— Generate a new private key/ public certificate. You will then keep the private key (file *.pem*) in a safe location ;
— Deploy the new certificate *.crt* on your clients in the folder `C:\Program Files (x86)\ssl` on Windows or `/opt/wapt/ssl` on Linux and MacOS manually, using a GPO or using an Ansible role (not documented) ;

63.2.2 Manipulation sur les paquets en conséquence

Les paquets du dépôt local étant signé avec l'ancienne clé, il convient de re-signer l'intégralité des paquets avec la nouvelle clé.

Afin de re-signer tous les paquets WAPT avec la nouvelle clé (les paquets logiciel et les paquets machine), utiliser la commande :

```
wapt-get sign-packages C:\\waptdev\\*
```

63.3 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 `C:\Program Files (x86)\wapt\ssl`.

Cette opération peut être effectuée à l'aide d'une GPO, manuellement, avec un paquet WAPT ou avec un rôle Ansible (non documenté).

63.4 Comment déplacer mon dépôt sur une autre partition

For any reason, you may need move the repository to another partition.

Your repository contains 3 folders which can be quite large :

— `wapt` ;
— `wapt-host` ;
— `waptwua` ;

Avertissement : Nous n'utiliserons pas la même méthode pour Linux et Windows.

63.4.1 Linux

Sous Linux, créez un point de montage sur `fstab`.

For this example, the second partition is named *part2*.

part2 is an **ext4 formated partition**.

Debian / Ubuntu

— Create a temporary folder.

```
mkdir /mnt/tmp
```

— Create a temporary mount point.

```
mount /dev/part2 /mnt/tmp
```

— Move the folders.

```
mv /var/www /mnt/tmp
```

— Unmont the partition.

```
umount /dev/part2
```

— Edit the `fstab` file.

```
vi /etc/fstab
```

— Add the following line to the `fstab` file.

```
# <file system> <mount point> <type>  <options> <dump>  <pass>
/dev/part2      /var/www      ext4    defaults  0       0
```

— Mount the partition.

```
mount -a
```

Indication : If there is no error, the partition is mounted.

— You can check by running.

```
df -h

#Result
Filesystem     1K-blocks     Used Available Use% Mounted on
dev/part2       15G          944M       14G   7% /var/www
```

— Remove the temporary folder.

```
rm -rf mnt/tmp
```

Centos / RedHat

— Create a temporary folder for copying the folders.

```
mkdir /mnt/tmp
```

— Create a temporary mount point.

```
mount /dev/part2 /mnt/tmp
```

— Move the folders.

```
mv /var/www/html /mnt/tmp
```

— Unmont the partition.

```
umount /dev/part2
```

— Edit the `fstab` file.

```
vi /etc/fstab
```

— Add the following line to the `fstab` file.

```
# <file system> <mount point> <type>  <options> <dump>  <pass>
/dev/part2      /var/www/html      ext4    defaults  0       0
```

— Mount the partition.

```
mount -a
```

Indication : If there is no error, the partition is mounted.

— You can check by running.

```
df -h

#Result
Filesystem     1K-blocks     Used Available Use% Mounted on
dev/part2       15G          944M       14G   7% /var/www
```

— Remove the temporary folder.

```
rm -rf mnt/tmp
```

Windows

Sous Windows, la meilleure méthode est de *sauvegarder* et *restaurer* le serveur sur la nouvelle partition.

Note : It is possible to install the server on another partition than `C:`.

63.5 Mon UUID BIOS bogue

— Some problems happen sometimes with some BIOSes. WAPT uses the *UUID* of the machine as the host identifier.
— The *UUID* is supposed to be unique. Unfortunately, for some OEMs (Original Equipment Manufacturers) and some manufacturing batches, BIOS *UUID* are identical.
— The machine will register in the WAPT console but it will replace an existing device, considering that the machine has only changed its name.

63.5.1 Résolution

WAPT permet de générer un *UUID* aléatoire pour remplacer celui indiqué dans le BIOS.

```
wapt-get generate-uuid
```

63.6 Mon WAPTdeploy ne fonctionne pas

63.6.1 Symptôme

L'utilitaire **waptdeploy** n'arrive pas à installer l'agent WAPT.

63.6.2 Résolution

Ajouter l'argument `waptsetupurl` dans les paramètres de la stratégie de déploiement **waptdeploy**.

```
--waptsetupurl=https://monserverserveurwapt/waptagent.exe
```

63.6.3 Lancer WAPTdeploy localement

Lancer **waptdeploy** localement peut mettre en évidence un problème en affichant explicitement les erreurs.

Exemple de commande à lancer :

```
C:\Program Files (x86)\wapt\waptdeploy.exe --
↪hash=2a9971aad083d6822b6e4d1ccfb9886be9429ec58bb13246810ff3d6a56ce887 --minversion=1.4.2.0 --
↪wait=15
```

Dans notre cas le hash n'est pas le bon.

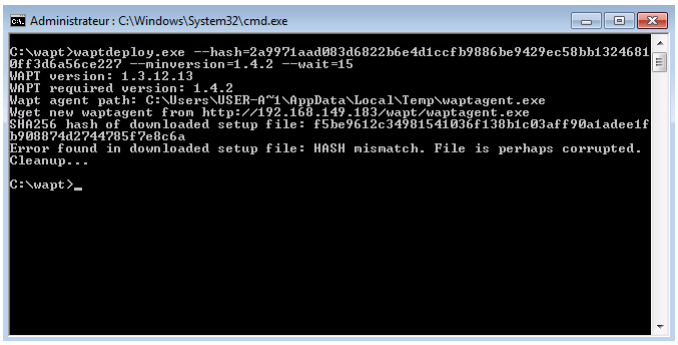


FIG. 1 – Error with WAPTDeploy

Attention : Ne pas oublier de lancer l'invite de commande en tant qu'*Administrateur*.

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
```

If **waptdeploy** does not show up you will have to double check the GPO settings :

 # You may be using an old waptdeploy version, then
téléchargez la dernière version de **waptdeploy** sur le store WAPT.

 # Thanks to Emmanuel EUGENE from French 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`.

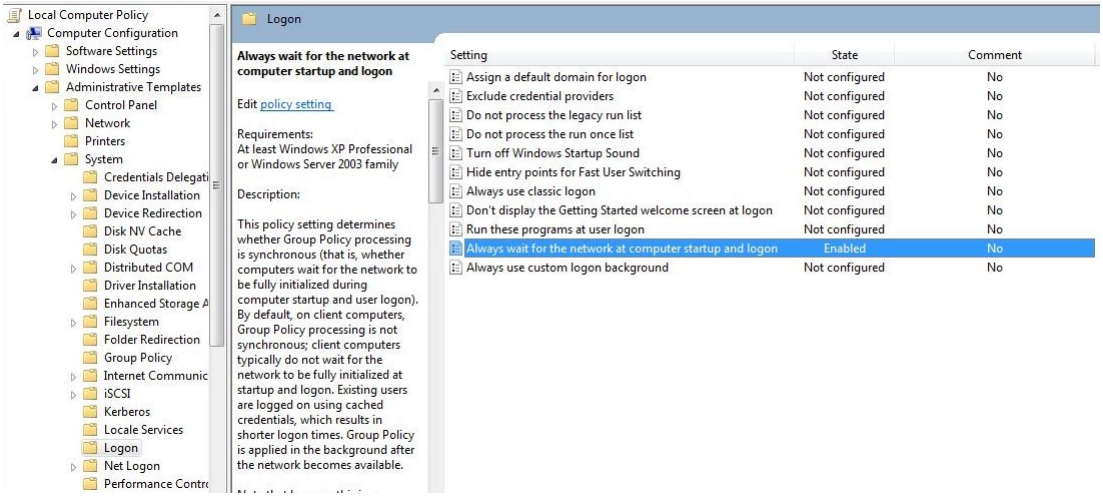
63.7 Windows n'attend pas le réseau au démarrage

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.

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



63.8 WAPTexit 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.

63.8.1 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*.

Un paquet WAPT existe pour traiter le problème :

— A WAPT package exists for this purpose : tis-disable-hybrid-shutdown.

63.8.2 Résolution : Windows Home édition

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`.

63.8.3 Résolution : GPO locale corrompue

Il peut arriver que les stratégies de groupes locales de la machine soient corrompues.

Une des solutions consiste à supprimer les stratégies locales actuelles en supprimant le fichier `C:\windows\system32\GroupPolicy\gpt.ini`, puis en redémarrant la machine, et enfin en relançant l'installation de la tâche d'extinction :

```
wapt-get add-upgrade-shutdown
```

Si le problème se reproduit, cela signifie peut être qu'une autre application manipule également la GPO.

63.9 WAPTexit 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.

63.9.1 Solution : augmenter le délai d'installation

Pour résoudre le soucis, il faut modifier la valeur `preshutdowntimeout` ainsi que la valeur `max_gpo_script_wait`.

Définissez ces valeurs dans `C:\Program Files (x86)\wapt\wapt-get.ini` pour modifier le comportement par défaut.

```
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`.

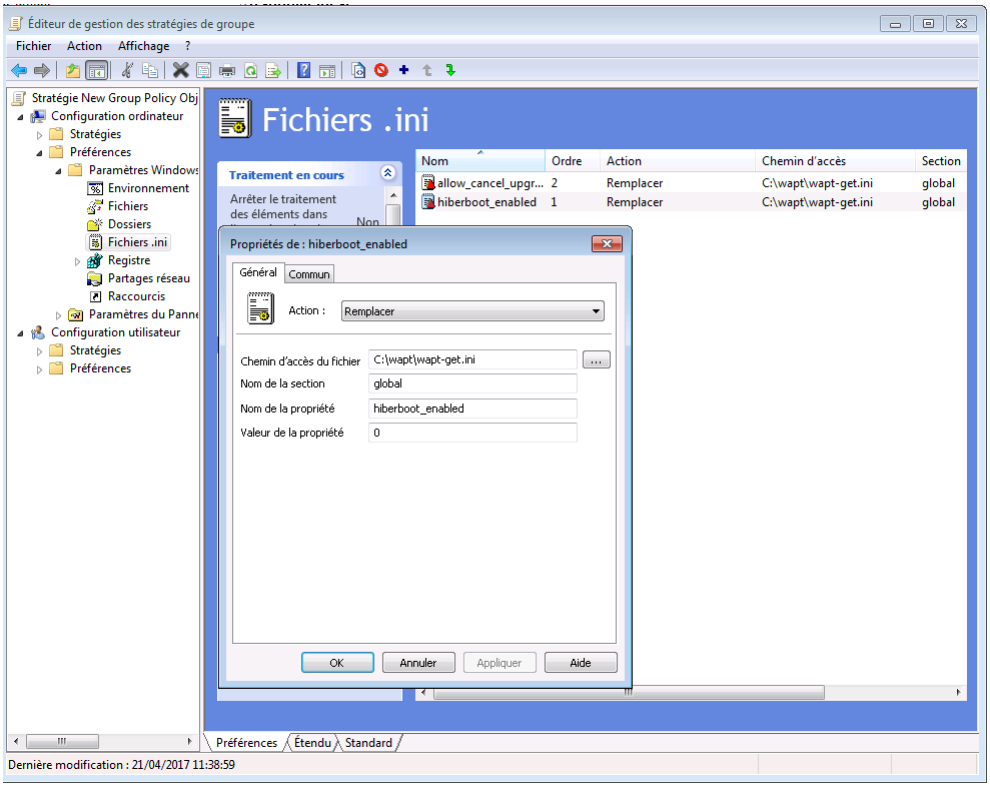


FIG. 2 – GPO ini File

63.10 Message d'erreur à l'ouverture de la console

63.10.1 Vérification de la version

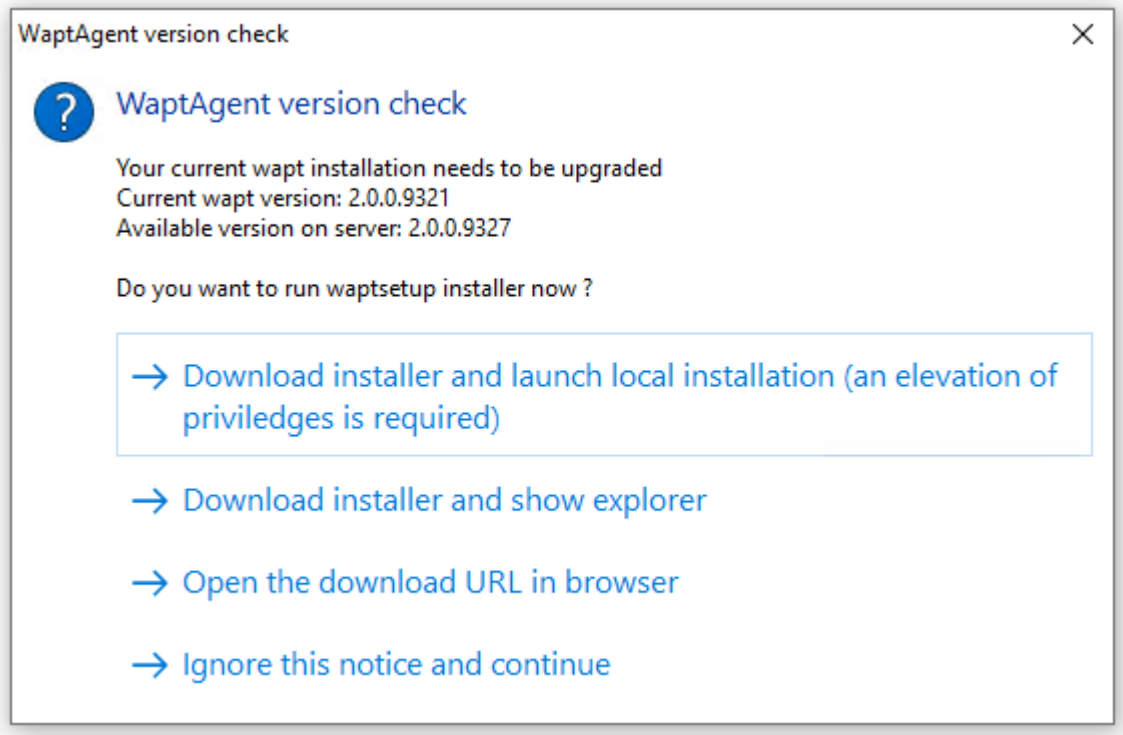


FIG. 3 – Version d'erreur de la console WAPT

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.

63.10.2 Connection refused

La console WAPT ne parvient pas à joindre le port 443 du serveur.

— Check whether the **Nginx** web service is running on the WAPT Server.

```
ps aux | grep nginx
```

— If **Nginx** is not running, restart the **Nginx** service.

```
service nginx restart
```

— If **Nginx** still does not start, you'll need to analyze journal logs in `/var/log/nginx/` on Linux or in `C:\Program Files (x86)\wapt\waptserver\nginx\logs` on Windows.

63.10.3 Service Unavaible

Il est possible que le service *waptserver* soit stoppé.

— Check whether **waptserver** is running.

```
ps aux | grep wapt
```

— If the command returns nothing, then start the **waptserver**.

```
service waptserver start
```

63.10.4 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`.

— Close your WAPT console.
— Delete the folder `%appdata%\..\Local\waptconsole`.
— Launch the command `wapt-get enable-check-certificate`.
— Be sure that the previous command has gone well.
— Restart the WAPT service with `net stop waptservice && net start waptservice`.
— Restart the WAPT console.

Dans le cas ou 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.

63.11 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.

63.12 Problème lors du enable-check-certificate

63.12.1 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 :

- Check the value of *wapt_server* in your `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 regénérer vos certificats autosignés.
- On the WAPT Server, delete the content of the `/opt/wapt/waptserver/apache/ssl/` folder.
  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.
- You may now retry **`enable-check-certificate`**.

63.13 Problème avec la création de paquet

63.13.1 Création de paquet via la console

Le glisser-déposer dans le dépôt privé dans la console WAPT ne fonctionne pas :

- The method will not work if the WAPT console has been started without *Local Administrator* privilege.
- The method will not work if the WAPT console has been started with UAC.
  Solution de contournement simple : passer par *Outils* → *Créer un modèle de paquet depuis un installeur* → *Choisir un installeur*.
- The WAPT console does not fill in automatically the informations in the fields :
  - There are special characters in some file path of the binary.
  - The installer does not provide the desired informations.

63.13.2 Problème de droit avec l'invite de commande Windows

Lors de l'édition d'un paquet, on obtient le message suivant :

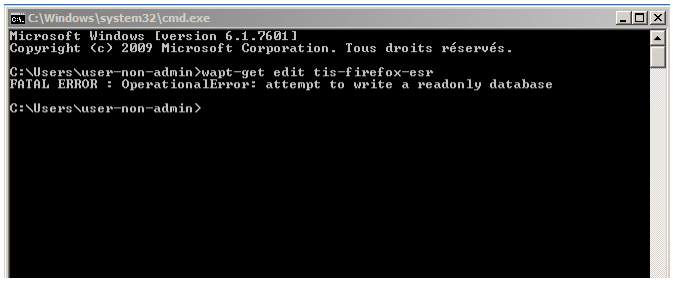


FIG. 4 – OperationnalError : attempt to write a readlony database

Solution

Ouvrir une session en tant qu'*Administrateur Local* et refaire l'opération souhaitée.

63.13.3 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 :

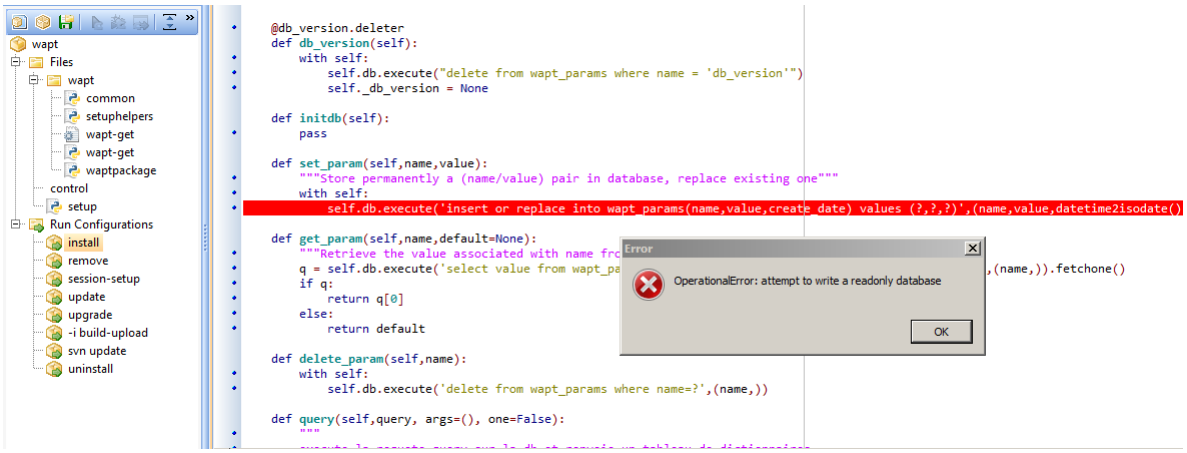


FIG. 5 – OperationnalError : attempt to write a readlony database

Solution

Ouvrir une session en tant qu'*Administrateur Local* et refaire l'opération souhaitée.

63.13.4 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

— Build the package with **PyScripter** or manually *build the package*.

Indication : Si l'**upload** a précédemment échoué, vous pourrez trouver le paquet généré dans `C:\waptdev`.

— Download and install **WinSCP** using WAPT.

```
wapt-get install tis-winscp
```

— Using **WinSCP**, **upload** your package in `/var/www/html/wapt/` path of you Linux server.
— Once the upload has finished, you'll need to recreate the `Packages` index file on your repository.

```
wapt-scanpackages /var/www/wapt/
```

63.13.5 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

— If WAPT returns an error code, research the error code on the 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.

— Go to the computer and try to install the package with the WAPT command line utility. Then check that the software has installed.

> **Attention :** Une fois l'installation silencieuse lancée, ne pas intervenir.
> L'objectif est de reproduire le comportement de l'agent WAPT.

— If the package installs silently in user context, this may mean that the software installer does not work in *SYSTEM* context.
— If it is still not working, launch the installation manually. It is possible for an error to appear explicitly describing the problem (ex : missing dependency, etc).
— It is possible that the installer does not support installing over an older version of the software, so you will have to explicitly remove older versions of the application before installing the new one.

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'installeur lui rende la main.

si l'installeur 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, augmentez 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 :

- It checks the presence of the *uninstall key* on the computer.
- It checks that the version number of the software is equal or greater than the version number in the `control` file.
- If this is not the case, it infers that the software may not be present on the computer.

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*..

- If the software has installed correctly, this may mean that the uninstall key or the software version in the package is not correct.
- Retrieve the correct *uninstall key* and make changes to the WAPT package accordingly.
- If the error happens when using the **install_msi_if_needed** function, this means that the MSI installer is badly packaged and that it is returning an incorrect *uninstall key*.

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 :

— Use the argument `get_version` to provide the path to another `uninstallkey`.

```
def install():

   def versnaps2(key):
       return key['name'].replace('NAPS2 ','')

   install_exe_if_needed('naps2-5.3.3-setup.exe',silentflags='/VERYSILENT',key='NAPS2 (Not Another␣
↪PDF Scanner 2)_is1',get_version=versnaps2)
```

— Providing an empty value for `min_version` tells WAPT not to check for versions.

```
min_version=' '
```

Attention : Avec cette méthode **on ne vérifiera plus la version lors de mise à jour !**

63.14 Problèmes fréquents liés aux Antivirus

Certains Antivirus lèvent des alertes pour des composants de WAPT.

Parmis 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"
```

63.15 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).

63.16 I have a lot more hosts in the console than I have host packages on my server ?

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.

63.17 Erreurs courantes

63.17.1 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 :

- A web server is extremely easy to setup, secure, maintain, backup and monitor.
- To work correctly, a WAPT package needs to be self-contained. Indeed, we do not know if the network will be available at the time of the installation launch (for example if we have a waptexit that starts when the workstation is shutting down on a network with 802.1x user authentication, there will no longer be a network available at the time of installation). The self-contained nature of WAPT makes it more deterministic than other deployment solutions.
- Network congestion may result from downloading large packages on large fleets of devices because you have less control over bandwidth rates or you may not be able to finish a partial download.
- This method breaks or at least weakens the security framework of WAPT.
- This method does not allow you to expose your repositories to Internet for your traveling personnel.

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**.

63.18 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**.

CHAPITRE 64

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.

You can explore the role source code by visiting Tranquil IT repository on Github.

64.1 Pré-requis

— hôtes Debian Linux ou CentOS ;
— un sudoer account sur ces machines ;
— Ansible 2.8.

64.2 Installer le rôle Ansible

64.2.1 Discovery

64.2.2 Enterprise

— Install `tranquilit.waptserver` Ansible role.

```
ansible-galaxy install tranquilit.waptserver
```

— To install the role elsewhere, use the *-p* subcommand like this.

```
ansible-galaxy install tranquilit.waptserver -p /path/to/role/directory/
```

64.3 Utiliser le rôle Ansible

— Ensure you have a working ssh key deployed on your hosts, if not you can generate and copy one like below.

```
ssh-keygen -t ed25519
ssh-copy-id -i id_ed25519.pub user@srvwapt.mydomain.lan
ssh user@srvwapt.mydomain.lan -i id_ed25519.pub
```

— Edit Ansible hosts inventory (`./hosts`) and add the Linux hosts.

```
[srvwapt]
srvwapt.mydomain.lan ansible_host=192.168.1.40
```

— Create a playbook with the following content in `./playbooks/wapt.yml`.

```
- hosts: srvwapt
  roles:
    - { role: tranquilit.waptserver }
```

— Run your playbook with the following command.

```
ansible-playbook -i ./hosts ./playbooks/wapt.yml -u user --become --become-method=sudo -K
```

The server is now ready. You may go to the documentation on *installing the WAPT console* ! !

64.3.1 Paramètres des rôles Ansible

Les variables disponibles sont énumérées ci-dessous, ainsi que les valeurs par défaut (voir `defaults/main.yml`) :

— Version of WAPT that will be installed from WAPT Deb/RPM repository.

```
wapt_version: "2.0"
```

— Version of PostgreSQL that will be installed from WAPT Deb/RPM repository.

```
pgsql_version: "11"
```

— Version of CentOS used for RPM repository address.

```
centos_version: "centos7"
```

— The parameter `launch_postconf` defaults to True, it launches WAPT Server post-configuration script silently.

```
launch_postconf: True
```

64.3.2 Exemple d'un *playbook* Ansible

Voici un exemple d'un *playbook* Ansible.

```
- hosts: srvwapt
  vars_files:
    - vars/main.yml
  roles:
    - tranquilit.waptserver
```

64.4 Deploying the Linux WAPT Agent with Ansible

To avoid mistakes and automate your WAPT agents deployment on Linux, we provide Ansible roles for installing WAPT agents on :
- Debian ;
- Ubuntu ;
- Redhat / CentOS.

You can explore the role source code by visiting this link on Github.

64.4.1 Pré-requis

- hôtes Debian Linux ou CentOS ;
- un sudoer account sur ces machines ;
- Ansible 2.8 ;

64.4.2 Installer le rôle Ansible

- Install `tranquilit.waptagent` Ansible role.

```
ansible-galaxy install tranquilit.waptagent
```

- To install the role elsewhere, use the *-p* subcommand like this.

```
ansible-galaxy install tranquilit.waptagent -p /path/to/role/directory/
```

64.4.3 Utiliser le rôle Ansible

- Ensure you have a working ssh key deployed on your hosts, if not you can generate and copy one like below.

```
ssh-keygen -t ed25519
ssh-copy-id -i id_ed25519.pub user@computer1.mydomain.lan
ssh user@computer1.mydomain.lan -i id_ed25519.pub
```

- Edit Ansible hosts inventory in the `./hosts` file and add the Linux hosts.

```
[computers]
computer1.mydomain.lan ansible_host=192.168.1.50
computer1.mydomain.lan ansible_host=192.168.1.60
```

- Create a playbook with the following content in `./playbooks/deploywaptagent.yml`.

```
- hosts: computers
  roles:
    - { role: tranquilit.waptagent }
```

- Ensure all variables are correctly set (see *wapt-get.ini variables*).
  - `wapt_server_url` ;
  - `wapt_repo_url` ;
  - `wapt_crt`.

Important : Variables configuration is important as it will configure the behavior of the WAPT.

You **must** replace the default certificate with your Code-Signing public certificate.

— Run your playbook with the following command.

```
ansible-playbook -i ./hosts ./playbooks/deploywaptagent.yml -u user --become --become-method=sudo -K
```

Congratulations, you have installed your WAPT agent on your Linux hosts!

64.4.4 Paramètres des rôles Ansible

Available variables are listed below, along with default values (see `defaults/main.yml`).

64.4.5 WAPT agent variables

— Version of WAPT that will be installed from WAPT Deb/RPM repository.

```
wapt_version: "2.0"
```

— Version of CentOS used for RPM repository address.

```
centos_version: "centos7"
```

wapt-get.ini variables

The `wapt_server_url` parameter points to your WAPT server and is used by default for the `wapt_repo_url`.

```
wapt_server_url: "https://srvwapt.mydomain.lan"
wapt_repo_url: "{{ wapt_server_url }}/wapt/"
```

You can override it like so :

```
wapt_server_url: "https://wapt.landomain.lan"
wapt_repo_url: "https://wapt.otherdomain.com/wapt/"
```

Certificate filename located in `files/` subdirectory of the role :

```
wapt_crt: "wapt_ca.crt"
```

64.4.6 Exemple d'un *playbook* Ansible

Voici un exemple d'un *playbook* Ansible.

```
- hosts: hosts
  vars_files:
    - vars/main.yml
  roles:
    - tranquilit.waptagent
```

CHAPITRE 65

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.

65.1 API V1

65.1.1 /api/v1/hosts

— Get registration data of one or several hosts.

```
# 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)

(suite de la page précédente)

```
# 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 = wgets('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 = wgets('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
```

Ceui-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
```

65.1.2 /api/v1/groups

— récupère tous les paquets groupes. Les groupes peuvent être trouvés avec la section *groupe* dans le paquet.

```
group_wapt = wgets('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

---

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 = wgets('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 = wgets('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 = wgets('https://%s:%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 = wgets('https://%s:%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
```

65.1.3 /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 =  wgets('https://%s:%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
```

65.2 API V2

65.2.1 /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
```

65.3 API V3

65.3.1 /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
```

65.3.2 /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
```

65.3.3 /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))
```

65.3.4 /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
```

65.3.5 /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
```

65.3.6 /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
```

65.3.7 /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.

65.3.8 /api/v3/upload_packages

À faire : Tests

65.3.9 /api/v3/upload_hosts

À faire : Tests

65.3.10 /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"
```

65.3.11 /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

65.3.12 /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"
```

65.3.13 /api/v3/reset_hosts_sid

Il y a plusieurs possibilités : 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}
```

65.3.14 /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}
```

65.3.15 /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}
```

65.3.16 /api/v3/trigger_host_action

À faire : Tests

65.3.17 /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"
```

65.3.18 /api/v3/ping

Ping shows a general set of informations on a WAPT Server.

```
# https://srvwapt.mydomain.lan/ping
# Lists WAPT Server informations

ping_wapt =  wgets('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))
```

CHAPITRE 66

Contacter l'éditeur de WAPT

Contactez-nous pour plus d'informations :

- **Tranquil IT** : https://www.tranquil.it/.
- **Twitter** : https://twitter.com/tranquil_it.
- **Linkedin** : https://www.linkedin.com/company/tranquil-it.
- **Forum in French** : https://forum.tranquil.it/.
- **Forum in English** : https://www.reddit.com/r/WAPT.

CHAPITRE 67

Glossaire

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.

Déployeur de Paquet
Déployeurs de Paquet
: Un **Déployeur 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 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** is a domain name that specifies its exact location in the tree hierarchy of the Domain Name System. It specifies all domain levels, including the top-level domain and the root zone. FQDN example : wapt.nantes.pdl.organization.fr.

EPEL
: **Extra Packages for Enterprise Linux** est un dépôt additionel 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** is the agent of WAPT installed on each computer client.

waptexit
: **waptexit** is a command will launch by Windows shutdown script (on Professional version). The command run standby upgrade packages.

waptsetup
: **waptsetup** is a setup to install a WAPT console.

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.

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.

CHAPITRE 68

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.

68.1 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 **wapttray**, **waptservice** et **waptexit** de l'agent WAPT peuvent être optionnellement désactivées en fonction du contexte d'usage.

68.2 Périmètre à sécuriser

Les éléments à sécuriser qui concernent strictement WAPT sont :

- **The WAPT Server** (*waptserver*).
- **The WAPT agents** (*wapt-get*) and its sub-components (*wapttray*, *waptservice* et *waptexit*).
- **The management console** (*waptconsole*).
- **The network communications** between these different components.

En complément des éléments listés ci-dessus, un exploitant de WAPT devra choisir et suivre une méthodologie adaptée au contexte de son *Organisation* pour :

- Insure the safe provisioning of all other files that are to be incorporated into a WAPT package.
- Develop the WAPT package python `setup.py` script so as to avoid any exploitable security or confidentiality defect.
- Manage in a safe way the private keys for signing the packages.
- Manage in a safe way the Authorities of Certification and Revocation for the SSL and HTTPS certificates.

La gestion sûre de ces éléments complémentaires est exclue du périmètre de cette documentation.

68.3 Description des utilisateurs typiques

Les rôles suivants doivent être compris pour évaluer les principes de sécurité présents dans WAPT :

- **Utilisateur**
  A *User* is an individual/ user of a WAPT equipped end-device (**Enterprise** and **Discovery**).
- **Déployeur de Paquet**
  A *Package Deployer* is an individual with the ability to sign packages that **DO NOT** contain python code (generally *group*, *host* and *unit* packages) and with the ability to upload the package to the main repository (**Enterprise**).
- **Développeur de Paquet**
  A *Package Developer* is an individual with the ability to sign any package, may it include or not include python code, and to upload the package to the main repository (**Enterprise**) ;

Note : The distinction between *Package Deployer* and *Package Developer* only exists in the **Enterprise** version of WAPT.

- **SuperAdmin**
  The *SuperAdmin* is an individual with all rights within WAPT.
- **Administrateur Local**
  A *Local Administrator* is an individual with local administration right of the WAPT equipped end-devices (**Enterprise** and **Discovery**) ;

Note : En fonction du contexte de la documentation et de la version du produit, un *Administrateur* désignera un *Déployeur 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.

68.4 Description of the sensitive assets in 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) :
- Availability ;
- Integrity ;
- Confidentiality ;
- Authenticity.

Les biens sensibles à protéger sont les suivants :

68.4.1 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
- Integrity ;
- Confidentiality ;
- Authenticity.

68.4.2 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
- Integrity ;
- Confidentiality.

68.4.3 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
- Availability.

68.4.4 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
— Integrity ;
— Confidentiality.

68.4.5 Bien sensible B5 : Les exécutables WAPT installés sur les postes client

Les exécutables 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
— Integrity.

68.4.6 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 protégées.

Note : Besoin de sécurité de l'authentification
— Integrity ;
— Confidentiality.

68.5 Description des hypothèses sur l'environment d'exploitation de WAPT

Par définition, les hypothèses sont des déclarations portant sur le contexte d'emploi de WAPT ou de son environment.

Les hypothèses suivantes sur l'environment d'exploitation de WAPT doivent être considérées :

68.5.1 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.

68.5.2 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.

68.5.3 Hypothèse H3 : Les binaires nécessaires au fonctionnement de WAPT sont intègres

Toutes les librairies 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.

68.5.4 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.

68.5.5 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.

68.5.6 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.

68.6 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 :

— **Unauthorized entities** : it is a human attacker or an entity that interacts with WAPT without legitimately having access to it.

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 :

68.6.1 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.

68.6.2 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.

68.6.3 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.

68.6.4 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.

68.6.5 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.

68.7 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.

68.7.1 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
- The registering of an illegitimate device in the database.
- A denial-of-service attack by overloading the database.
- The insertion of a fraudulent inventory in the database.

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) :
- The sending of sensitive informations to an illegitimate and unauthorized WAPT Server.
- The recovery of sensitive informations by an unauthorized entity.
- The display of fake information in the management console of the *Administrator*.
- An incorrect date to be sent upon a HEAD request, thus preventing future upgrades (request for a modified file date).
- Sending the WAPT console password to an illegitimate and unauthorized WAPT Server.

Solution mise en place

Pour fonctionner correctement en version sécurisée :
- An option for verifying the server HTTPS certificate is introduced in `C:\Program Files (x86)\waptwapt-get.ini` on the WAPT agents that will **force the verification of the server certificate by the WAPT agents**.
- An option for verifying the server HTTPS certificate is introduced in `C:\Program Files (x86)\waptwapt-get.ini` on the WAPT agents that will force the verification of the server certificate by the **WAPT console**.

L'implémentation technique peut être basée sur deux méthodes :
- By using a certificate verification tool implemented in the configuration file of WAPT's **Nginx** web server ; this method is typically provided by a *Certificate Authority* that is trusted by your network.
- By using the *certificate pinning* method, which consists of providing the WAPT agent a short list of trusted certificates that will be stored in `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
- An unauthorized entity using an open port fraudulently.

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
- An unauthorized entity sending a fake inventory for a device that rightfully exists in the database.

Solution mise en place

- On the first **register**, each device generates a key/ certificate pair that is stored in `C:\Program Files (x86)\wapt\private`, only accessible in read-only mode to *Local Administrators*. Once the device has successfully registered, the public key is sent to the WAPT Server.
- When the inventory is updated, the new inventory status is sent along with the private key of the device. The new inventory is then decrypted with the public key stored in the database.
- The server will refuse any inventory that is signed with a wrong key.

Fonction de sécurité F1E : Vérification des droits avant l'exécution de certaines actions WAPT

Note : risques traités
- Avoid the execution of sensitive tasks on WAPT clients by unauthorized entities.

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` (update the available list of packages).
- `wapt-get upgrade` (launch waiting upgrades).
- `wapt-get download-upgrade` (download waiting upgrades).
- `wapt-get clean` (remove packages left in cache after installation).
- stop any running WAPT task.
- stop/ reload the WAPT service.

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` : requests the WAPT agent to install a WAPT package flagged as **MISSING**.
— `wapt-get remove` : requests the WAPT agent to remove a package.
— `wapt-get forget` : requests the WAPT agent to forget the existence of a previously installed WAPT package without removing the software or the configuration.

68.7.2 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
— To avoid an unauthorized entity modifying the content or the behavior of a WAPT package.

Solution mise en place

— When an *Administrator* or a *Package Deployer* builds a WAPT package, the file `manifest.sha256` is created that lists the control sums of all files in the package.
— A file `signature.sha256` **encrypted** with the WAPT agent's private key is then created in the folder `WAPT`; it contains the control sum of the file `manifest.sha256`.
— The whole is then compressed and suffixed with a *`.wapt`* extension.
— When a WAPT agent downloads a WAPT package, the agent checks that the file `signature.sha256` has been signed with the private key that matches the certificate present in the folder `WAPT`.
— The WAPT agent then checks that the certificate or the chain of certificates in `certificate.crt` has been signed with a key matching one of the certificates present in the folder `C:\Program Files (x86)\wapt\ssl`.
— The WAPT agent then generates the control sum of all the files contained in the package (except the files `signature.sha256` and `certificate.crt`) and verifies that it matches the file `manifest.sha256` contained in the package.
— If one of these steps does not pass, this means that a file has been modified/ added/ removed. The execution of the `setup.py` is then canceled.
— The altered package is then deleted from the local cache and the event is journalized in the logs of the 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
— An unauthorized entity modifying WAPT dependencies on WAPT equipped devices by falsifying `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-méme 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
- An unauthorized entity modifying the behavior of a WAPT agent.

Le répertoire d'installation `C:\Program Files (x86)\wapt` est accessible en lecture et modification :
- To the *Local Administrators* by direct access to the installation folder of the WAPT agent on the device.
- To the *Administrators* through the deployment of WAPT agent upgrades.

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
- An unauthorized entity falsifying an inventory status update.
- An unauthorized entity impersonating the identity of a WAPT equipped device.

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.

68.7.3 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
- An unauthorized entity sending falsified requests to the WAPT agents.

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` : requests the WAPT agent to install a WAPT package flagged as **MISSING**.
- `wapt-get remove` : requests the WAPT agent to remove a package.
- `wapt-get forget` : requests the WAPT agent to forget the existence of a previously installed WAPT package without removing the software or the configuration.
- `wapt-get update-status` : requests the WAPT agent to send its current inventory status to the WAPT Server.
- `wapt-get upgrade` : requests the WAPT agent to execute a package flagged as **NEED UPGRADE**.
- `wapt-get update` : requests the WAPT agent to update the list of available packages.

L'ensemble des attributs de cette requête sont signés :
- The device's *UUID* ;
- The action (ex : **`install`**) ;
- The arguments (ex : tis-firefox) ;
- The timestamp of the requests.

Le certificate associé à la signature est également passé :
- Upon receiving a request, the WAPT agent verifies that the request has been properly signed.
- The agent will the verify that the timestamp is within a one minute delay window.
- Ultimately, the agent will verify that the certificate is authorized to launch actions.

68.8 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

Les processus cryptographiques sont utilisés dans les activités suivantes :

- Signature and verification of the **files contained in a package**.
- Signature and verification of the **attributes of a package**.
- Signature and verification of **instantaneous actions** on the WAPT clients.
- Signature of inventories and **status of WAPT clients**.
- Authentication of the WAPT client Websocket connections on the WAPT server.
- HTTPS communication between the WAPT clients and the WAPT Server.
- HTTPS communication between the WAPT console and the WAPT Server.
- HTTPS communication between the WAPT clients and the WAPT repositories.

68.8.1 Répertoires et fichiers référencés dans ce document

- <WAPT> : WAPT installation folder. By default `%Program Files (x86)%WAPT`.
- <WAPT>`wapt-get.ini` : WAPT client configuration file (**`wapt-get`** and **`waptservice`**).
- <WAPT>`ssl` : default directory for signed actions and trusted certificates.
- <WAPT>`sslserver` : default directory for storing HTTPS server certificates (pinning).
- <WAPT>`private` : default certificate directory for signing the inventory and the Websocket connections.
- `%LOCALAPPDATA%waptconsolewaptconsole.ini` : configuration file for the console and package development actions for the **`wapt-get`** tool.
- `%appdata%waptconsolessl` : default trusted certificate directory for importing packages from an external repository (i.e. *package templates*).

68.8.2 Definitions of Actors

— **Organisation**
An Organization is the realm of responsibility within which WAPT is used.
— **Autorité de Certification**
A Certificate Authority is the entity that keeps the keys that have been used to sign certificates for the *Package Developers*, the *Package Deployers* and the HTTPS servers.
— **Administrateurs**
Administrators have a personal RSA key and a certificate that has been signed by the *Certificate Authority* of the *Organization* ; they also have a login and a password for accessing the WAPT console.
— **Postes clients WAPT**
WAPT clients are the realm of devices that the *Organization* has allowed the *Administrators* to manage with WAPT. The clients **may or may not be a member** of the *Organization*'s Active Directory domain.
— **Serveur WAPT**
The WAPT Server is the Linux / Nginx/ PostgreSQL that the *Organization* uses to keep the inventory of WAPT equipped devices.
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**
Internal WAPT repositories are one or several Linux/ Nginx servers that deliver signed WAPT packages to WAPT clients using the HTTPS protocol.
— **External WAPT repositories**
External WAPT repositories are a public WAPT repository that the *Package Developers* may use to import packages designed by other *Organizations*, under the condition that they check the adequacy of the WAPT package in regards the internal policies on security and safety ;
— **Serveur Active Directory**
The Active Directory Server manages the *Organization*'s AD domain ;

68.8.3 Synthèse des modules crypto mis en oeuvre par la solution WAPT

À faire : sfonteneau

Coté client WAPT (WAPT 1.5.0.12) :
— **Python 2.7.13** standard *ssl* module linked on **`OpenSSL 1.0.2j 26 Sep 2016`** : used for the HTTPS connections between the WAPT clients and the WAPT server.
— **cryptography==1.9** linked on **`openssl 1.1.0f`** : used for all RSA crypto operations such as key generations, X509 certificate generations and signature verifications.
— **kerberos-sspi==0.2** and **requests-kerberos==0.11.0** : used for authenticating the WAPT client on its first registering on the WAPT Server.
— **pyOpenSSL==17.0.0** : used to recover the WAPT Server certificate chain.
— **certifi==2017.4.17** : used as base for the Root Authorities certificates.
— **Openssl 1.0.2l** dll : used in waptcommon.pas written with the FPC Indy library and the TIdSSLIOHandlerSocketOpenSSL class.
Coté serveur WAPT :
— **nginx/1.10.2** : configured for TLS1.2, cipher "EECDH+AESGCM :EDH+AESGCM :AES256+EECDH :AES256+EDH".
— **python 2.7.5** standard *ssl* module linked on **`OpenSSL 1.0.1e-fips 11 Feb 2013`**.
— **cryptography==1.9** linked on **`OpenSSL 1.0.1e-fips 11 Feb 2013`** : used for all RSA crypto operations such as key generations, X509 certificate generations and signature verifications.

68.8.4 Key and certificate management for the Administrators

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 :
- A private 2048 bit *RSA* key that has been encrypted by the aes-256-cbc algorithm.
- A *X509* certificate signed by an *Certificate Authority* trusted by the *Organization*.

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 :
- The generated RSA key is 2048bit long, encrypted with aes-256-cbc, encoded in PEM format and saved with a `.pem` extension.
- The certificate is either self-signed, or signed by a Trusted Authority from whom we have received a key and a PEM formated certificate.
- If the certificate is self-signed, then its *KeyUsage* attribute contains the *keyCertSign* flag.
- If the *Administrator* is authorized by the *Organization* to sign packages that contain python code (the presence of a `setup.py` file is detected in the package), the *extendedKeyUsage* attribute of the certificate contains the **CodeSigning** flag.
- The *X509* certificate is encoded and handed over to the *Administrator* in PEM format with a `.crt` extension.

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 vérifcation 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** >= *notValidBefore* et **Now** <= *notValidAfter*).

Authorizing an Administrator's certificate to sign a package

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%waptconsolewaptconsole.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*.

68.8.5 Managing the WAPT agent's key and certificate

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 :
- When updating the WAPT client status on the server (**`update_server_status`**) **signing informations**.
- When the WAPT agent establishes a Websocket with the server (**`waptservice`**) **signing the WAPT client UUID**.

Émission initiale et mise à jour du certificat du client WAPT

— On finishing the installation process of the WAPT agent on the device, the WAPT agent automatically registers itself on the WAPT Server by sending a kerberos authenticated HTTPS request that uses the TGT of the machine account.

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.

If the key and the certificates have not yet been generated, or if they do not match the current *FQDN* of the device, the WAPT agent generates a self-signed RSA key and X509 certificate with the following parameters :

— The key is 2048 bit RSA encoded in PEM format and stored in the file <WAPT>private<device FQDN>.pem.
— The generated certificate has the following attributes :
  — *Subject.COMMON_NAME* = <device FQDN>.
  — *Subject.ORGANIZATIONAL_UNIT_NAME* = name of the *Organization* registered in the WAPT client's Windows registry.
  — *SubjectAlternativeName.DNSName* = <device FQDN>.
  — *BasicConstraint.CA* = True.
  — *Validity* = 10 years.
  — *Serialnumber* = random ;
— The certificate is saved in the <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.

— The inventory and the WAPT agent status updates are sent to the WAPT Server over POST HTTPS requests ;
— The POST HTTPS requests are authenticated by adding two specific headers :

— *X-Signature* :
  — JSON encoded BLOB of inventory or status informations.
  — signature of the JSON file with the private key of the WAPT Client : *sha256* hashing and *PKCS#1 v1.5* padding.
  — *base64* encoding of the signature.
— *X-Signer* : *Subject.COMMON_NAME* ou UUID du client WAPT.

— After having initially authenticated the WAPT client with kerberos, the WAPT Server receives the certificate sent by the Client and stores it in the table *hosts* of its inventory in PEM format (column *host_certificate*).

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

PEM formatted certificates are stored in files with *.crt* or *.pem* extensions in the directory defined with the public_certs_dir parameter in the <WAPT>wapt-get.ini file. They are reputed to be **trusted certificates**.

The public_certs_dir parameter is initialized by default to be <WAPT>ssl.

Le déploiement de ces certificats d'autorité est effectué lors de l'installation initiale de l'agent WAPT par l'installeur.

Depuis la console, l'*Administrateur* compile un installeur personnalisé en vue de son déploiement par *GPO* sur les Postes clients.

La console WAPT incorpore dans cet installeur les certificats présents dans le répertoire <WAPT>\ssl du poste depuis lequel l'installeur 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
  Vérifie le certificat du serveur https en utilisant le bundle fourni par le module **certifi**. 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'installeur de l'agent WAPT, les certificats ou le bundle de certificats sont intégrés dans l'installeur.

Lors du déploiement de l'installeur 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.

68.8.6 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 un connexion chiffrée avec le protocole TLS et utilise côté client le même bundle de certificat que la connexion HTTPS Client vers Serveur WAPT.

68.8.7 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
  Vérifie le certificat du serveur https en utilisant le bundle fourni par le module **certifi**. 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`** ;

68.8.8 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>`.
- shell command `wapt-signpackage.py <WAPT-package-list>`.
- saving a *host* package in the WAPT console.
- saving a *group* package in the WAPT console.
- importing a package from an external repository.
- creating a package with the MSI setup wizard.

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)* ;

Signing the attributes in the control file

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.

Etapes du processus :

- The attributes *signed_attributes*, *signer*, *signature_date*, *signer_fingerprint* are added to the structure of the `control` file :
  - *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* : hexadecimal encoded `sha256` fingerprint of the fingerprint obtained with the **`fingerprint`** function included in the **`cryptography.x509.Certificate`** class.
- The attributes of the control structure are JSON encoded.
- The resulting JSON BLOB is signed with *sha256* hashing and *PKCS#1 v1.5* padding.
- The signature is base64 encoded and stored as a *signature* attribute in the `control` file.

Signature des fichiers du paquet

- The `control` file attributes are signed and serialized in JSON format. The result is stored in the <WAPT>`control` file of the WAPT package.
- The PEM serialized X509 certificate of the certificate holder is stored in the <WAPT>`certificate.crt` file of the WAPT package.
- The *sha256* fingerprints of the all files contained in the WAPT package are hexadecimal encoded and stored as a JSON list [(filename,hash),] in the <WAPT>`manifest.sha256` file in the WAPT package.
- The content of the file <WAPT>`manifest.sha256` is signed with the private key of the *Administrator* (2048 bit RAS key), *sha256* hashed and *PKCS#1 v1.5* padded :
  - The signature process relies on the signing function of the **`cryptography.rsa.RSAPrivateKey.signer`** class.
  - **`cryptography.rsa.RSAPrivateKey.signer`** relies on the OpenSSL functions of **`EVP_DigestSignInit`**.
- The signature is base64 serialized and stored in the file <WAPT>`signature.sha256` of the WAPT package.

68.8.9 Vérifier la signature des attributs d'un paquet

Elle a lieu :

- When the index file of available packages on the WAPT client is updated from the `Packages` index file on the repository.
- When a package signature is verified (installation, download) when not in *development* mode, i.e. if the installation takes place from a ZIP file and not from a development directory.

Elle consiste à :

- Reading the control attributes from the WAPT package's `control` file.
- Recovering the X509 certificate from the certificate holder from the WAPT package's `certificate.crt` file.
- Decoding the base64 formated signature attribute.
- Constructing a JSON structure with the attributes to be signed (such as defined in the **`PackageEntry`** class).
- Verifying if the public key of the holder's certificate can verify the hash of the JSON structured list of attributes and the signature of the `control` file, using *sha256* hashing and *PKCS#1 v1.5* padding.
- Verifying whether the certificate is trusted (either it is present in the list of trusted certificates, or signed by a Trusted *Certificate Authority*).

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 formated certificate fingerprint>.crt`.

Un attribut de la structure `control` du paquet indique l'empreinte du certificat du signataire du fichier `control`.

68.8.10 Vérifier la signature d'un paquet

Elle a lieu :
- When installing a package on a WAPT client.
- When editing an existing package.
- When importing a package from an external repository (if the checkbox is checked in the WAPT console).

Elle consiste à :
- Recovering the X509 certificate from the certificate holder from the WAPT package's `certificate.crt` file.
- Verifying that the certificate has been signed by a Trusted Authority whose certificate is present in the folder `ssl` on the WAPT client.
- Verifying the signature of the file `manifest.sha256` with the public key.

68.8.11 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

- The action is defined by its name and the actions attributes. The attributes are *uuid*, *action*, *force*, *notify_server*, and *packages* (for actions implicating a list of packages).
- The attributes *signed_attributes*, *signer*, *signature_date*, *signer_certificate* are added to the structure of the action :
  - *signed_attributes* list of the attributes that are signed.
  - *signer* Subject.COMMON_NAME of certificate holder.
  - *signature_date* : current date and time (UTC) in "%Y-%m-%dT%H :%M :%S" format.
  - *signer_certificate* certificate holder's base64 encoded *X509* certificate.
- The structure is JSON encoded.
- The signature of the JSON file is calculated from the RSA private key of the *signer* using a *sha256* hash algorithm and a *PKCS1 v1.5* padding.
- The signature is base64 encoded and stored on the *signature* attribute inside the JSON file.

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 :
- The X509 certificate of the certificate holder is extracted from the JSON file (format PEM).
- The WAPT client tests whether the certificate is to be trusted, i.e. that it is present in <WAPT>ssl or that it has been signed by a Trusted Authority (certificate of the Authority present in <WAPT>ssl).
- The WAPT client checks whether the certificate can verify the signature that is present in the JSON structure of the action, which consists of :
  - Extracting the base64 encoded signature from the *signature* attribute in the JSON file ;
  - Extracting the signature date formated in "%Y-%m-%dT%H :%M :%S" from the *signature_date* attribute ;
  - Checking that the signature date is neither too old in the past, nor too late into the future by over 10 minutes ;
  - Reconstructing a JSON representation of the attributes of the action ;
  - Checking that the certificate's public key can verify the JSON file with the signature by using a *sha256* hash algorithm and a *PKCS1 v1.5* padding.

68.8.12 Verifying the complete download of a package

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.

This md5 sum is only used to insure that a package has been fully downloaded.

The checking of the signature of the package will be used instead of the md5 sum to fully insure the integrity and the authenticity of the WAPT package.

CHAPITRE 69

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.

69.1 Variables d'environnement

— Les variables d'environnement existent depuis le DOS. Elles peuvent vous faciliter la vie (et la mienne).

69.2 Répertoires des programmes

— Not every system uses `C:\` as the main drive. Some enterprises use folder redirection, and relocate the Documents folder. Some places in the world do not speak English and their directories reflect that. **Use those environmental variables to make your programs just work** :
  — `%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.

69.3 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.

69.4 Codes d'erreur

— Utilisez les codes d'erreur standard lorsque vous quittez votre programme.

69.5 Impression

— Utilisez l'"API d'impression Windows et n'utilisez pas l'impression directe dans votre programme.

69.6 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**.

69.7 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.

69.8 Numéros de version

— Utilisez le versionnage semantique (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** !

69.9 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**.

69.10 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 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*.

69.11 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 pares-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 !

69.12 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.

CHAPITRE 70

Stratégie de sortie des mises à jour de WAPT

WAPT does not release on a fixed date schedule.

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.

70.1 Délai de publication entre les versions Enterprise et Discovery

A new major version will be released as **Enterprise** and that same version will be released as an **Discovery** RC1 (Release Candidate #1). Before releasing, the Enterprise version will have undergone thorough internal testing and testing with valued **Insider Program** customers to insure no regression has slipped into the core of WAPT.

The Enterprise release will cycle through several RCs and the final general availability Enterprise version will then become available between 4 and 8 weeks after the first release to our **Insider Program** customers.

This delay will provide several benefits to the General Availability release process :

- It allows additional time to perform in depth testing of the new Enterprise features while avoiding major regressions.
- It allows Tranquil IT to work with a small set of selected Enterprise customers to insure upgrade procedures work smoothly.
- It gives sufficient time to the forum and the mailing list to index questions and answers to eventually include into the official documentation.
- It gives Tranquil IT's documentation team a fixed functional target to document the new or improved features.
- It gives the translation team the necessary delay to update translations.
- It gives the communication and marketing team a fixed functional target and a capacity to backward schedule announcements, video podcasts and overall promotion.

CHAPITRE 71

Security bulletin

71.1 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.

Parts (H1) : ################### with overline Chapters (H2) : ********** with overline Sections (H3) : ========== Subsections (H4) : —————— Subsubsections (H5) : ^^^^^^^^^^^^^^^^^^ Paragraphs (H6) : « « » » » » » » » » » » » » » » » » » » »

description
Changelog

keywords
WAPT, Histoire, Genèse, documentation

CHAPITRE 72

Changelog

72.1 WAPT-2.0 Serie

72.1.1 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)

72.1.2 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.

72.1.3 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.

72.1.4 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.tmpl 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

72.1.5 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

72.1.6 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

72.2 WAPT 1.8 Serie

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

72.2.1 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

72.2.2 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.

72.2.3 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

72.2.4 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 ExceptionLogger. 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.

72.2.5 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.

72.2.6 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).

72.2.7 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.

72.2.8 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 *setuphelpers* 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 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] 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.

72.2.9 WAPT-1.8.2.7265 RC2 (2020-05-29)

hash git : 339f1996

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

72.2.10 WAPT-1.8.2.7165 RC1 (2020-05-29)

hash git : 1387b38f

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 lastest 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 setuphelpers 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 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 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.

72.2.11 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 ;

72.2.12 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 ;

72.2.13 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 uploding 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 cient 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 ;

72.2.14 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** ;

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 *socle* 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**;

72.3 WAPT 1.7 and older

72.3.1 WAPT-1.7.4.6078 (2019-05-17)

(hash 95a146c002)
— [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;

72.3.2 WAPT-1.7.4-6183 (2019-09-09)

(hash da870a2c)
— [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;

72.3.3 WAPT-1.7.4-6183 (2019-09-09)

(hash2090b0e6d52cecfb04f8fa4c279e7c0a0252d6e2
— [FIX] **wapt-get session-setup** : fix bad print in **session_setup**.
Regression introduced in b30b1b1a550a4 (1.7.4.6229);

72.3.4 WAPT-1.7.4-6230 (2019-10-23) (not released)

(hash da870a2c)
— [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`;

72.3.5 WAPT-1.7.4.6082 (2019-05-20)

(hash 5b6851ae)
— [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) ;

72.3.6 WAPT-1.7.4.6143 (2019-06-25)

(hash da870a2c)
— [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 ;

72.3.7 WAPT-1.7.4.6165 (2019-08-02)

(hash f153fab4)
— [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`;

72.3.8 WAPT-1.7.4-6183 (2019-09-09)

(hash 38e08433)
— [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 ;

72.3.9 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 ?) ;

72.3.10 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 checkupgrades.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 ;

72.3.11 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 checkupgrades.json.
Was not reflecting an up to date state.
— [IMP] waptself :

— early support of high DPI ;
— loading of icons in the background ;

72.3.12 WAPT-1.7.4.6078 (2019-05-17)

(hash 5b6851ae)

Fixes

— [FIX] takes *profile* packages (AD based groups)

into account (**Enterprise** only)

72.3.13 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) ;

72.3.14 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] setuphelpers `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)

72.3.15 WAPT-1.7.4 RC2 (2019-04-30)

(hash 5ef3487)

Security

— upgrade **`urllib3`** to 1.24.2 for

CVE-2019-11324 (high severity) ;
— upgrade **`jinja2`** 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 ;

72.3.16 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 *uwaptpollthreads* 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 ;

72.3.17 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 wapttray 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 unproperly 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 reproductible 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 ;

72.3.18 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%waptconsolewaptconsole.ini [global] default_ca_key_path setting);
— option /CACertFilename : path to CA certificate to use for signing
the new certificate (defaults to %LOCALAPPDATA%waptconsolewaptconsole.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%waptconsolewaptconsole.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 cryptopgraphy 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` ;

72.3.19 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;

72.3.20 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;

72.3.21 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 waptdb (**Enterprise** only) ;

Improvements

— [ADD] versioning on group packages filenames ;
— [ADD] button to create AD Host profiles

(package automatically installed/removed based on AD Grouo 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 ;

72.3.22 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 popupmenu commands which are launched

for the selection of hosts. Custom variables {uid} ;
— Other improvements :

— [IMP] French translations fixes ;

72.3.23 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 ;

72.3.24 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 skipped ;

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.

72.3.25 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 ;

72.3.26 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;

72.3.27 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 `wsusscan2` 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 `wsusscan2` cab file from Microsoft (daemon/ service wapttasks) ;
— 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 **posgresql** 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** ;

72.3.28 WAPT-1.6.2.8 (2018-10-09)

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 wapttasks 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.

72.3.29 WAPT-1.6.1.0 (2018-06-21)

Fixes

— [FIX] av potential cause in wapttray ;
— [IMP] buffer LogOuput ;
— [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] wapttray 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] wapttray : 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 wapttray ;
— [FIX] no default `audit_period` ;
— [REMOVED] **`zeromq`**, replaced by long http polling between wapttray,

wapt-get and waptservice ;

72.3.30 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 ;

72.3.31 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 ;

72.3.32 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=sslserversrvwapt.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 dispkay 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 ;

72.3.33 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.

72.3.34 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

WAPTupgrade 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`.

72.3.35 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

72.3.36 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` ;
— Add handling of 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 ;

72.3.37 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 fill 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 (segragation 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 PotsgreSQL 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 ;

72.3.38 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
(simplication 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)

72.3.39 WAPT-1.5.1.14 (2018-01-09)

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 differenciated 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;

setuphelpers

— *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 identitiy 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

wapttray ;
— 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 dépots 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 ;

72.3.40 WAPT-1.4.0 (2017-05-05)

— pas de release officielle ;
— [NEW] migration sur la base PostgreSQL à la place de MongoDB ;

72.3.41 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

72.3.42 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/fr/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 dépot.

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 errreur ;

— [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.tmpl`
— [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) ;

72.3.43 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.

72.3.44 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 poping 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 ;

72.3.45 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"

72.3.46 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_statictics if mongodb lacks aggregate support
— [IMP] register host on server in postconf using **`waptservice`** http
instead of command line **`wapt-get`**

72.3.47 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**.

72.3.48 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 unecessary 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 >= 2.4

72.3.49 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 ;

Watpserver

— [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 **setuphelpers** 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://monserveurwapt:8080 with *wapt_server* = https://monserveurwapt.

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:\wapt\waptserver\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 advized 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 command :*update* / command :*upgrade* / command :*remove* / command :*forget* / command :*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` ;

72.3.50 WAPT-1.0 (2015-01-31)

— [ADD] first public version of WAPT

CHAPITRE 73

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.

73.1 Définitions

Dans le présent accord, les termes en majuscules ont la signification suivante :

- "AFFILIATE" means any entity that directly or indirectly, through one or more intermediaries, controls, is controlled by, or is under common control with the subject entity. For purposes of this definition, "control" means direct or indirect possession of the power to direct or cause the direction of the management and policies of an entity, whether through the ownership of voting securities, by contract or otherwise.
- "AUTHORIZED PURPOSES" means Customer's internal business purposes.
- "CUSTOMER SYSTEM" means Customer's internal website(s), servers and other equipment and software, including, without limitation, mobile devices and systems based on virtual or logical emulations.
- "DELIVERY DATE" means the date, set forth in the applicable, on which the Software is scheduled to be made available to Customer.
- "DOCUMENTATION" means the printed, paper, electronic or online user instructions and help files made generally available by Company for use with the Software, as may be updated from time to time by Company.
- "INTELLECTUAL PROPERTY RIGHTS" means all intellectual property rights or similar proprietary rights, including (a) patent rights and utility models, (b) copyrights and database rights, (c) trademarks, trade names, domain names and trade dress and the goodwill associated therewith, (d) trade secrets, (e) mask works, and (f) industrial design rights ; in each case, including any registrations of, applications to register, and renewals and extensions of, any of the foregoing in any jurisdiction in the world.

— "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).
— "SALES CONFIRMATION" means the ordering documents for Services and licenses for Software purchased from Company that are entered into or otherwise mutually accepted in writing hereunder by the parties from time to time, including modifications, supplements and addenda thereto. If there is any inconsistency or conflict between a Sales Confirmation and this Agreement, this Agreement controls, unless the Sales Confirmation specifically identifies by Section reference the provision that such Sales Confirmation is modifying, and then such change will apply for such Sales Confirmation only. Any terms or conditions stated in any Sales Confirmation, sales acknowledgment or invoice (except for details of the software provided, price, quantity of licenses, subscription term, Delivery Date and other details of delivery which are not pre-printed and which are inconsistent with the terms of this Agreement) shall be of no force and effect, and no course of dealing, usage of trade, or course of performance shall be relevant to explain or modify any term expressed in this Agreement.
— "SOFTWARE" means the WAPT Enterprise and WAPT packages specified in a Sales Confirmation and any Company Updates that Company provides to Customer in accordance with Support Services that Customer is entitled to receive pursuant to this Agreement, in object code form or in text form. For all purposes of this Agreement, "Software" excludes any open source software and all Third Party Offerings, such as third party software, content and other virtual and digital assets.
— "SUPPORT SERVICES" means the support and maintenance services provided by Company under Section *Services de maintenance et de support* below.
— "THIRD PARTY OFFERINGS" means certain software or services delivered or performed by third parties that are required for the operation of the Software.
— "UPDATES" means bug fixes, patches and maintenance releases to the Software to the extent made generally available by Company to its licensees.
— "DEVICES" means any of Customer's or its Affiliates' device on which the WAPT Enterprise agent is installed.

73.2 Commandes, octroi de licence, clé d'activation et garantie de performance

73.2.1 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.

73.2.2 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.

73.2.3 Restrictions

Vous ne pouvez pas, directement ou indirectement, et vous ne permettrez à aucun utilisateur ou tiers de :

— Reverse engineer, decompile, disassemble or otherwise attempt to discover the source code or underlying ideas or algorithms of the Software.
— Modify, translate, or create derivative works based on any element of the Software or any related Documentation.
— Rent, lease, distribute, sell, resell, assign, or otherwise transfer your rights to use the Software.
— Use the Software for timesharing purposes or otherwise for the benefit of any person or entity other than for the benefit of Customer.
— Remove any proprietary notices from the Documentation.
— Publish or disclose to third parties any evaluation or benchmarking of the Software without Company's prior written consent.
— Use the Software for any purpose other than its intended purpose.
— Interfere with or disrupt the integrity or performance of the Software ; or
— Attempt to gain unauthorized access to the Software or Company's systems or networks.

73.2.4 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.

73.2.5 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.

73.3 Confidentialité

73.3.1 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, sketches, drawings, models, inventions (whether or not patented or patentable), know-how, processes, apparatus, formulae, equipment, algorithms, software programs, software source documents, APIs, and other creative works (whether or not copyrighted or copyrightable).
— Information concerning research, experimental work, development, design details and specifications, engineering, financial information, procurement requirements, purchasing, manufacturing, customer lists, business forecasts, sales and merchandising and marketing plans and information.
— Proprietary or confidential information of any third party who may disclose such information to Disclosing Party or Receiving Party in the course of Disclosing Party's business ; and

— The terms of this Agreement and any Sales Confirmation or SOW. Confidential Information of Company shall include the Software, the documentation, the pricing, and information regarding the characteristics, features or performance of Beta Licenses and Non-GA Solutions. Confidential Information also includes all summaries and abstracts of Confidential Information.

73.3.2 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.

73.3.3 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 :

— Was known by the Receiving Party prior to receipt from the Disclosing Party either itself or through receipt directly or indirectly from a source other than one having an obligation of confidentiality to the Disclosing Party.
— Was developed by the Receiving Party without use of the Disclosing Party's Confidential Information ; or
— Becomes publicly known or otherwise ceases to be secret or confidential, except as a result of a breach of this Agreement or any obligation of confidentiality by the Receiving Party. Nothing in this Agreement shall prevent the Receiving Party from disclosing Confidential Information to the extent the Receiving Party is legally compelled to do so by any governmental investigative or judicial agency pursuant to proceedings over which such agency has jurisdiction ; provided, however, that prior to any such disclosure, the Receiving Party shall :
  — Assert the confidential nature of the Confidential Information to the agency.
  — Immediately notify the Disclosing Party in writing of the agency's order or request to disclose ; and (z) cooperate fully with the Disclosing Party in protecting against any such disclosure and in obtaining a protective order narrowing the scope of the compelled disclosure and protecting its confidentiality.

73.3.4 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.

73.4 Droits de propriété

73.4.1 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.

73.4.2 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.

73.4.3 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.

73.5 Obligations du Client

Le Client est responsable de :

— Obtaining, deploying and maintaining Customer's computer system and all computer hardware, software, peripherals and network equipment necessary for Customer, its Affiliates and their respective users to use the Software ; and

— Paying all third party fees and access charges incurred in connection with the foregoing. Except as specifically set forth in this Agreement or a Sales Confirmation, Company shall not be responsible for supplying any hardware or other equipment to Customer under this Agreement.

73.6 Services de maintenance et de support

73.6.1 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 :

— Technical support by forum and mailing list (in a reasonable time frame) if Enterprise Support has not been subscribed.
— Technical support by phone if Enterprise Support has been subscribed and.
— 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.

73.6.2 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 »).

73.7 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.

73.8 Déclarations et garanties ; Clause de non-responsabilité

73.8.1 Représentations et garanties mutuelles

Chaque partie déclare, garantit et s'engage à ce que :

— It has the full power and authority to enter into this Agreement and to perform its obligations hereunder, without the need for any consents, approvals or immunities not yet obtained ; and
— Its acceptance of and performance under this Agreement shall not breach any oral or written agreement with any third party or any obligation owed by it to any third party to keep any information or materials in confidence or in trust.

73.8.2 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.

73.9 Indemnisation ; limitation de la responsabilité

73.9.1 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 :

— Obtain for Customer the right to use the Software.
— Substitute a functionality equivalent, non-infringing replacement for such the Software.
— Modify the Software to make it non-infringing and functionally equivalent ; or
— Terminate this Agreement.

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 :

- Use of the Software in combination with any software, hardware, network or system not supplied by Company where the alleged infringement relates to such combination.
- Any modification or alteration of the Software other than by Company.
- Customer's continued use of the Software after Company notifies Customer to discontinue use because of an infringement claim.
- Use of Open Source Software.
- la violation par le client du droit applicable ; et
- Customer System.

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.

73.9.2 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 :

- Customer's use or alleged use of the Software other than as permitted under this Agreement.
- Customer or its Affiliates' users use of the Software in violation of any applicable law or regulation, or the Intellectual Property Rights or other rights of any third part ; or
- Arising from the occurrence of any of the exclusions set forth in Section *Exclusions*. Customer shall pay all damages, costs and expenses, including attorneys' fees and costs (whether by settlement or award of by a final judicial judgment) paid to the Third Party bringing any such Third-Party Claim. Customer's obligations under this Section *Indemnisation des Clients* are conditioned upon :
  - Customer being promptly notified in writing of any claim under this 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.

73.9.3 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.

73.10 Durée et résiliation

73.10.1 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.

73.10.2 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 :

— Files a petition for bankruptcy or has a petition for bankruptcy filed against it that is not dismissed within sixty (60) days after filing or admits its inability to pay its debts as they mature, makes an assignment for the benefit of its creditors or ceases to function as a going concern or to conduct its operations in the normal course of business and such termination shall occur immediately upon notice ; or

— Commits a material breach of any provision of this Agreement and does not remedy such breach within 30 days (or 10 days after a failure to pay any fees hereunder) after receipt of notice from the other party or such other period as the parties may agree. Upon any termination for cause by Company, Customer shall pay any unpaid fees covering the remainder of the term of all Sales Confirmations after the effective date of termination. In no event shall any termination relieve Customer of the obligation to pay any fees payable to Company for the period prior to the effective date of termination.

73.10.3 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.

73.10.4 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 :

— Return to the Disclosing Party, or at the Disclosing Party's option, the Receiving Party shall destroy, all items of Confidential Information then in the Receiving Party's possession or control, including any copies, extracts or portions thereof ; and
— Upon request shall certify in writing to Disclosing Party that it has complied with the foregoing.

73.11 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.

73.12 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.

73.13 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.

73.14 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.

73.15 Droit applicable ; juridiction et lieu de juridiction

Dans toute réclamation découlant du présent accord ou s'y rapportant :

— The substantially prevailing party will recover its reasonable costs and attorneys' fees.
— Company and you each consent to the jurisdiction of the tribunal of Nantes (France) and waive any objection based on jurisdiction r venue, including forum non conveniens ; provided, however, if Company seeks injunctive relief, it may file such action wherever in its judgment relief might most effectively be obtained.
— The laws of France will apply, without regard to the choice of law provisions thereof.

73.16 Divers

73.16.1 Notices

Company may give notice to Customer by means of electronic mail to Customer's e-mail address on record with Company, or by written communication sent by first class postage prepaid mail or nationally recognized overnight delivery service to Customer's address on record with Company. Customer may give notice to Company by written communication sent by email to commercialtis@tranquil.it or by first class postage prepaid mail or nationally recognized overnight delivery service addressed to Tranquil IT, 12 avenue Jules Verne, 44230 Saint Sébastien sur Loire (France). Notice shall be deemed to have been given upon receipt or, if earlier, two (2) business days after mailing, as applicable. All communications and notices to be made or given pursuant to this Agreement shall be in the English or French language.

73.16.2 Clients du gouvernement américain

If Customer is a Federal United States Government entity, Company provides the Software, including related software and technology, for ultimate Federal Government end use solely in accordance with the following : Government technical data rights include only those rights customarily provided to the public with a commercial item or process and Government software rights related to the Software include only those rights customarily provided to the public, as defined in this Agreement. The technical data rights and customary commercial software license is provided in accordance with FAR 12.211 (Technical Data) and FAR 12.212 (Software) and, for Department of Defense transactions, DFAR 252.227-7015 (Technical Data – Commercial Items) and DFAR 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation). If greater rights are needed, a mutually acceptable written addendum specifically conveying such rights must be included in this Agreement.

73.16.3 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 CFRPart 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.
— The European Union and its member states, including without limitation Council Regulation (EC) No. 1334/2000.
— And other countries (collectively, "Export/Import Law"). Customer agrees to comply with all Export/Import Law applicable to the Software and assumes sole responsibility for obtaining licenses and other authorizations that are required under Export/Import Law to deliver and use the Software Material. The Licensee acknowledges and agrees and shall procure that any person to whom the Software is made available shall acknowledge and agree that the Software shall not be exported, re-exported or otherwise transferred to any countries for which the United States and/or the European Union maintains an embargo (collectively, "Embargoed Countries"), or a national or resident thereof, or to any person or entity on the U.S. Department of Treasury List of Specially Designated Nationals or the U.S. Department of Commerce Denied Parties List or Entity List, or to any person

on any comparable list maintained by the European Union or its member states (collectively, "Designated Nationals"). The lists of Embargoed Countries and Designated Nationals are subject to change without notice. Customer represents and warrants that neither it nor any of the persons to whom the Software is made available is located in, a national or resident of, or under the control of an Embargoed Country or Designated National. Customer specifically shall obtain all required authorizations from the U.S. Government before transferring or otherwise disclosing technical data or technology (as those term are defined in 22 CFR Part 120.10 and 15 CFR Part 722, respectively), to any Foreign Person (as defined in 22 CFR Part 120.16).

73.16.4 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.

CHAPITRE 74

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
Python	Python Software License
Librairies Python	Licenses OpenSource diverses
Lazarus	GNU Public Licence
Composants Lazarus	GNU Lesser General Public License
Librairies Lazarus	Licenses OpenSource diverses
OpenSSL	Openssl License
Redistr. Microsoft Visual C++	Microsoft Software License Terms
PostgreSQL	PostgreSQL License
NSSM	Public Domain
Nginx	2-clause BSD-like license