

Méthode d'installation automatisée de Windows (mode « unattended »)

Maurice Libes

Centre d'Océanologie de Marseille UMS 2196 CNRS
Campus de Luminy 13288 Marseille cedex 9
maurice.libes@univmed.fr

Xavier Nicolay

Faculté de Pharmacie de Marseille / CISCAM
27 Bd Jean Moulin, 13005 Marseille
xavier.nicolay@univmed.fr

Benjamin Gabaud

IUT-GTR Université de la Méditerranée
Campus de Luminy 13288 Marseille cedex 9
benjamin.gabaud@laposte.net

Résumé

Nous décrivons la technique d'installation de systèmes Windows dite "unattended", c'est à dire sans surveillance ni interaction avec l'administrateur.

Cette technique est basée sur la constitution d'un disque « source » Windows de base (le « master »), indépendant du matériel car il contient tous les drivers nécessaires (réseau, vidéos, son, cpu etc..) ainsi que toutes les mises à jour de sécurité du moment, qui ont été intégrés par la technique de « slipstreaming ». Cette version modifiée de Windows enrichie et préparée sur mesure, permet de couvrir les installations d'une grande majorité de matériels différents.

Un fichier de réponses préétablies « winnt.sif » ou « unattend.txt » permet alors d'automatiser entièrement l'installation de Windows. Cette automatisation fait gagner considérablement de temps aux administrateurs en réduisant leur présence devant le poste à installer et en enchaînant la post-installation des logiciels additionnels. Elle permet en outre une industrialisation des installations.

Dans une gestion de parc de PC, la technique « unattended » peut être utilisée comme méthode d'installation ou de déploiement complémentaire aux systèmes de clonage traditionnels, et facilite la maintenance générale du parc complet. Nous montrons dans cet article comment préparer un master Windows, le personnaliser, et deux méthodes d'installation automatique de ce « master » :

- la première en préparant un master windows qui est ensuite gravé sur CD

- la seconde en préparant un master windows disponible sur un partage cifs samba, par un boot réseau. Nous utilisons pour cela le projet « openSource » Unattended de SourceForge <http://unattended.sourceforge.net/>

Mots clefs

unattended, installation automatique, WindowsXP, gestion de parc, administration système, clonage, slipstreaming

1 Introduction - objectifs

La technique d'installation de systèmes Windows dite "unattended", c'est à dire sans surveillance, ou encore sans aucune interaction avec l'utilisateur, n'est en soi ni nouvelle ni novatrice. Elle est en effet prévue par Microsoft depuis longtemps, mais il nous semble qu'elle est peu connue et peu employée dans nos organismes. Le but recherché est d'automatiser complètement l'installation de Windows et d'obtenir des machines clones installées à l'identique. Aucune présence de l'administrateur n'est nécessaire jusqu'à la fin de l'installation. On gagne en définitive beaucoup de temps dans notre métier d'ASR.

Les objectifs seconds recherchés sont les suivants :

- partitionner ou gérer les partitions existantes sur le PC à installer en début de l'installation automatisée
- personnaliser l'installation de windows (installation ou enlèvement de tel ou tel composant de Windows, jointure au domaine Windows, etc ...),
- intégrer les mises à jour de sécurité (« hotfixes ») de Windows, l'objectif étant que le système soit à jour de tous les correctifs de sécurité avant le début de l'installation,
- intégrer une grande part de « pilotes » nécessaires à windows pour répondre à la grande diversité des matériels ciblés,
- rajouter automatiquement des logiciels additionnels en post-installation,
- avoir un système proposant des classes d'installation types (exemple: une classe d'installation pour des chercheurs, une autre pour des administratifs...),
- mettre en conformité le poste installé dans le parc informatique : affectation automatique d'un nom netbios, d'une adresse IP, intégration au domaine Windows, création des utilisateurs, d'un administrateur local du poste.

Dans les méthodes que nous avons utilisé, nous avons suivi des étapes de préparation sensiblement similaires :

- **La première étape** consiste à intégrer les mises à jour de sécurité de Windows « hotfixes » dans le master Windows.

- **La deuxième étape** consiste à intégrer différentes classes de pilotes (réseau, vidéo, chipsets, cpu, disques etc...), dans le master Windows.

- **La troisième étape** consiste à personnaliser le master Windows selon les besoins propres à l'ASR en enlevant certains composants du système jugés inutiles.

- **La quatrième étape** consiste à pouvoir enchaîner la post-installation en rajoutant automatiquement une liste de logiciels additionnels au système Windows.

2 Principes de base et pré-requis des installation en mode « unattended »

L'installation automatique de Windows repose sur l'utilisation d'un fichier de réponses préétabli appelé par défaut « *winnt.sif* » (pour une installation via cdrom) ou « *unattend.txt* » (pour une installation réseau). Les réponses inscrites dans ce fichier fournies au programme d'installation Windows permettent à l'administrateur système de ne rien avoir à répondre au clavier.

L'installation de Windows débute par l'exécution du programme « *winnt.exe* ». Sans argument, le fichier *winnt.sif* est lu (s'il existe), mais l'utilisation du switch */U* permet de spécifier un fichier de réponses alternatif comme « *unattend.txt* ». Le switch */S* spécifie l'emplacement des sources du « master » d'installation de Windows, disponibles sur CDROM ou sur un partage CIFS.

Exemple: *winnt.exe /U:unattend.txt /s:%CIFS%*:

2.1 Extrait d'un fichier de réponses « unattend.txt »

Le fichier *unattend.txt* est composé de plusieurs sections entre crochets, lesquelles comportent plusieurs couples *attribut=valeur du système*.

[Unattended]

Unattendmode = FullUnattended

[UserData]

ProductKey = aaaaaa-bbbbbb-cccccc-ddddd-eeee

OrgName = monlaboratoire

[GuiUnattended]

TimeZone = 105

[Identification]

DomainAdmin = mondomaine\nom_user

DomainAdminPassword = monpasswd

JoinDomain = mondomaine

...etc

On trouvera sur le Web des exemples assez complets de fichiers *unattend.txt* avec les différents sections et et les différents paramètres commentés sur les références suivantes :

- <http://www.winattended.com/personalisationwinxp.php>

- <http://www.winattended.com/unattend2.php>

Une option du logiciel « nLite » [Réf. 1], ou encore le programme *setupmgr.exe* de Windows présent dans le répertoire *C:\support\tools\deploy.cab* du CD d'installation de Windows permettent également de générer un fichier « *unattend.txt* ».

2.2 Structure du media d'installation de Windows, et temps clés des phases d'installation

2.2.1 Structure du media d'installation

Dans ce qui suit, le terme « master Windows » fera référence aux sources du système Windows copiées à partir du CDROM d'installation de Windows XP, vers le disque d'un PC. Dans le master Windows, le répertoire I386 (*D:\CDXP\I386*) contient les sources d'installation binaires de Windows et l'arborescence complète des fichiers d'installation.

- Au même niveau que le répertoire I386, Microsoft a prévu un mécanisme pour personnaliser l'installation. Il est possible de créer un répertoire \$OEM\$ qui aura un rôle particulier lors de l'installation. Ce répertoire peut contenir des utilitaires complémentaires à installer dans le système, tels que des pilotes matériels ou des applications supplémentaires. Si le setup de Windows trouve ce répertoire \$OEM\$, son contenu sera copié lors de l'installation sur le disque C: du PC à installer selon les règles suivantes :

- les éléments du dossier \$OEM\$\\$1 seront copiés dans C:

- les éléments du dossier \$OEM\$\\$2 seront copiés dans D:

- les éléments du dossier \$OEM\$\\$\$ seront copiés dans %systemroot%. (généralement *C:\System32*) lors de l'installation

- Ce répertoire \$OEM\$ peut en outre contenir un fichier appelé « *cmdlines.txt* » qui sera exécuté au temps T-12 minutes avant la fin de l'installation [Réf. 2]. Pour Windows ce moment privilégié T-12 impacte les installations relatives à tous les utilisateurs (All users).

On peut alors profiter de ce fichier pour installer certaines applications ou les derniers hotfix du moment qui n'ont pas été « slipstreamés ».

2.2.2 Les temps clés de l'installation Windows

Le processus d'installation est composé de plusieurs phases dans lesquelles il est possible d'intervenir afin d'installer des éléments ou modifier le système. Les phases clés de l'installation sont décrites sur <http://igm.univ-mlv.fr/~dr/XPOSE2004/jgers/time.html>

Les phases qui nous intéressent particulièrement sont celles où l'on va pouvoir intervenir pour personnaliser et finaliser l'installation à l'aide de fichier de commandes.

On retiendra plus particulièrement [Réf. 3] :

- à T-12 exécution de *cmdlines.txt* qui nous permettra de lancer l'installation de hotfixes additionnels par exemple

- à la fin de l'installation, lancement des commandes contenues dans la section [*GuiRunOnce*] du fichier

unattend.txt. La section [*GuiRunOnce*] contient les applications lancées au premier login du premier utilisateur dans l'ordre d'exécution.

3 Préparation d'un « master » d'installation

On commencera donc par copier le CD d'installation complet de Windows XP SP2 sur un disque D:XPSP2 par exemple. Toute la configuration ultérieure sera effectuée sur cette copie.

3.1 Intégration des hotfixes

Plusieurs méthodes plus ou moins bonnes ont été testées pour intégrer les hotfix au PC à installer. Selon nous la meilleure en terme de sécurité et d'efficacité est incontestablement celle du « *slipstreaming* » puisqu'elle modifie directement les sources du master d'installation.

Le « *slipstreaming* » est le processus consistant à intégrer un correctif de sécurité ou une mise à jour logicielle directement dans les sources d'installation de Windows. Cela revient à remplacer les fichiers originaux par les versions corrigées de ceux-ci directement dans les sources Windows sans que le numéro de version de Windows en soit affecté. L'installation de Windows subséquente est donc réalisée avec un « master » déjà à jour des correctifs de sécurité. On y gagne en sécurité puisque le système est déjà sécurisé dès le début de l'installation, et en durée d'installation puisque le PC n'a pas à aller chercher ses updates après coup, ni à rebooter après l'exécution de certains hotfix. Ce processus de *slipstreaming* peut se faire par le lancement des hotfixes en ligne de commande avec l'option */integrate* ou avec certains logiciels comme nLite.

3.1.1 Récupération des mises à jour de sécurité

Différents outils sont disponibles pour récupérer les hotfix. Nous avons utilisé WUD « Windows Update Downloader » [Réf. 4], mais également le script « *prepare* » de la distribution unattended de SourceForge (décrit plus bas), qui les récupère dans un répertoire spécial appelé « *updates* ».

3.1.2 Intégration des hotfix en slipstreaming par nLite

nLite est un programme « freeware » permettant de travailler directement sur les sources d'une installation de Windows [Réf. 1]. Une fois les hotfix récupérés par WUD, nous utilisons nLite pour intégrer les correctifs directement dans le master Windows. L'intégration est très simple : on indique à nLite le répertoire où est stocké le « master » Windows [D:XPSP2], on désigne ensuite le répertoire où ont été stockés les hotfixes que l'on a récupéré, et il suffit de cliquer sur l'option « *Hotfixes and Update pack* » pour appliquer le processus d'intégration des mises à jour sur le master Window.

3.1.3 Intégration des hotfix en slipstreaming par fichier de commandes

On peut également intégrer les hotfix par slipstreaming en ligne de commande avec l'option */integrate*. Il suffit alors

d'enchaîner les différents hotfix avec un fichier de commandes .cmd comme suit :

```
start /wait WindowsXP-KB931836-x86-FRA.exe /integrate:D:XPSP2
```

```
start /wait windowsxp-kb935448-x86-fra.exe /integrate:D:XPSP2
```

...etc.

3.1.4 Méthode complémentaire : intégration des hotfix avec cmdline.txt

Pour éviter des intégrations de hotfix en « *slipstreaming* » trop fréquentes dans le master Windows, nous pouvons également utiliser une méthode complémentaire d'intégration des hotfix grâce au fichier « *cmdlines.txt* » utilisé par Windows au temps T-12 avant la fin de l'installation. Cette méthode présente toutefois l'inconvénient de provoquer de temps en temps le redémarrage du PC selon la nature du hotfix qui est installé.

Le fichier « *cmdlines.txt* » contenu dans le répertoire I386/\$OEM\$, doit commencer par une ligne comportant le label de section [*Commands*]. Pour intégrer des hotfix complémentaires, la méthode est la suivante :

- création des répertoires `\OEM\$1\hotfix` sur le master Windows.

- dépôt des derniers hotfix, ou mises à jour de sécurité que nous voulons appliquer lors de l'installation, dans le répertoire `\OEM\$1\hotfix` de la distribution

- création du fichier `\OEM\$1\hotfix.bat` contenant le nom du hotfix à exécuter

- création du fichier `\OEM\cmdlines.txt` contenant
[Commands]
c:\hotfix.bat

Au temps T-12 de l'installation, le fichier *cmdlines.txt* est lancé automatiquement par l'installateur Windows. Il exécutera donc le contenu de *hotfix.bat*, qui applique les hotfix contenus dans c:\hotfix

```
c:\hotfix\KB896423.exe /passive /quiet /norestart
```

```
c:\hotfix\KB929969.exe /passive /quiet /norestart
```

Cette méthode est décrite sur le site Winattended [Réf. 2].

3.2 Allègement des sources de Windows par nLite

Outre le choix d'intégrer les hotfixes, nLite propose de nombreuses autres fonctionnalités. Nous avons utilisé plus particulièrement le choix « *remove* » pour enlever du master Windows des composants que nous jugeons inutiles (certains claviers ou langues étrangers) ou encore des programmes et jeux que nous ne voulons pas installer (IIS, outlook express etc...). On allège donc le volume des sources d'installation au maximum. Cette personnalisation de Windows permet une installation plus rapide en produisant un système d'exploitation plus léger et moins gourmand en ressources. Notons que dans le cas d'une installation automatique du « master » gravé sur CD cette

étape est capitale puisqu'il faut réduire le volume du master sur CD en dessous de 700M.

nLite propose également le choix d'intégrer des pilotes matériel dans le master Windows, et de générer le fichier « *unattend.txt* » mais nous avons réalisé ces étapes d'une autre manière.

3.3 Intégration de différentes classes de « pilotes » dans le master Windows

Dans le processus d'automatisation de l'installation d'un PC, le but est de n'être jamais ou le moins souvent possible, bloqué par l'absence d'un pilote de carte réseau, vidéo, de disque, de chipset... etc, qui ne serait pas présent dans les sources du système Windows. On a donc intérêt à enrichir au maximum le master Windows d'une grande quantité de drivers supplémentaires.

En effet, par défaut, Windows n'installe que ses propres drivers et applications telles que Internet Explorer, Windows Media player etc. Cependant, Microsoft donne la possibilité de copier d'autres fichiers pilotes lors de l'installation et donc de les intégrer directement dans le master d'installation de Windows.

Dans la section [Unattended] du fichier de réponses *unattended.txt* on n'oubliera pas de mettre

```
DriverSigningPolicy = Ignore
```

afin de ne pas être bloqué si un pilote sans signature numérique est présent.

- Récupération des pilotes Windows

Nous récupérons un grande diversité de pilotes matériels sur le site DriverPack [Réf. 5]. Ce site fournit de nombreux pilotes classés par rubriques : Chipset, DriverPack CPU, Graphics, LAN, MassStorage, Sound etc.. Le site propose également un intégrateur de drivers Dps_BASE.exe qui lui même permet d'utiliser deux méthodes d'intégration des pilotes dans le master Windows.

- **La première méthode d'installation des drivers** : consiste à décompresser tous les pilotes, et à les copier vers le répertoire \$OEM\$/1 dans un ensemble de répertoires désignés par des lettres correspondant aux différentes classes de drivers. Avec cette méthode, pour que Windows utilise les pilotes lors de l'installation, il faut, dans la section [unattend] du fichier « *unattend.txt* », renseigner l'attribut *OemPnpDriverPath* avec comme valeur la liste complète de tous les chemins de répertoires contenant les pilotes, comme suit :

```
OemPnpDriverPath
```

```
= \D\C\AM\1; \D\C\AM\2; \D\C\AM\3;\D\C\AM\4;\D\C\AM\5;\D\C\AM\6;\D\C\AM\7 ;\D\C\AT;\D\C\AU;\D\C\A\1;\D\C\A\2 ...etc...
```

Le problème majeur rencontré avec cette méthode est que la variable *OemPnpDriverPath* est limitée à 1024 caractères par Microsoft. Or la chaîne concaténée des chemins de répertoires contenant les pilotes dépasse très

largement cette valeur de 1024. Il se peut donc que certains pilotes se trouvant au delà du 1024ème caractère ne soient pas pris en compte par Windows.

Une manière de contourner cette limitation est d'utiliser le fichier \I386\HIVESFT.INF qui contient un ensemble de clés de registre, dont le registre *DevicePath* permettant de définir les chemins des pilotes, sans limitation de longueur: comme suit :

```
HKLM,"SOFTWARE\Microsoft\Windows\CurrentVersion",  
,"DevicePath",0x00020002,"%SystemRoot%\inf;  
%SystemDrive%\D\C\AM\1;%SystemDrive  
%\D\C\AM\2 ...;
```

Cette méthode a le mérite de fonctionner, mais est assez laborieuse à mettre en oeuvre (édition du fichier et syntaxe complexe à respecter) dans le cas de mise à jour de pack de drivers.

- La deuxième méthode d'installation des drivers

Cette méthode appelée Pyron (du nom de son auteur) [Réf. 6] crée un répertoire OEM (à ne pas confondre avec \$OEM\$ comme dans la méthode précédente) à la racine du CD d'installation (donc dans C:\OEM), répertoire dans lequel sont placés les ensembles de drivers compressés en 7zip.

La méthode modifie quelques fichiers du système et utilise un fichier *presetup.cmd* qui sera exécuté avant le setup de Windows. Lors de l'installation, le répertoire OEM contenant les drivers est copié et décompressé à la racine du disque dans C:\D

Le *presetup* exécute un programme *DevPath.exe* qui va chercher tous les chemins menant aux différents drivers (fichiers *.INF) dans C:\D et désactive la vérification de signature des drivers avec *DSPdsblr.exe*. La liste des sous répertoires menant aux pilotes est alors écrite dans

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\  
DevicePath. DevPath.exe fouille récursivement tous les  
répertoires contenant les drivers.
```

Nous avons retenu cette méthode car elle ne pose pas de problème de longueur de chemin menant aux drivers comme c'est le cas avec la méthode reposant sur l'attribut *OemPnpDriverPath*

4 Les méthodes de déploiement

4.1 Le projet Unattended de SourceForge

Cette distribution 100% opensource est disponible sur le site <http://unattended.sourceforge.net/>. Le principe de base de cette distribution est de faire démarrer le PC client à installer sur un noyau Linux (ou sur un système DOS). Ce noyau est paramétré pour monter par CIFS le répertoire d'une machine serveur « Unattended » qui lui fournit le master du système Windows à installer, ainsi qu'un ensemble de répertoires de travail. Un ensemble de scripts Perl enchaîne différentes phases de l'installation.

4.1.1 Coté serveur

La distribution « unattended » est installée sur un PC qui fait office de serveur du « master » Windows. Ce serveur, de nom netbios « *ntinstall* » doit avoir un service CIFS (Active Directory, ou samba pour ce qui nous concerne) correctement paramétré pour exporter le point de partage: `\\ntinstall\install` qui contient un ensemble de répertoires de travail, parmi lesquels :

dosbin: utilitaires pour DOS, et le script principal d'installation *install.pl* qui est exécuté quand le PC client démarre

site: Dans le répertoire `~/site/` un fichier « *unattend.txt* » préétabli contient les paramètres par défaut. Ces paramètres peuvent être changés interactivement par un script *install.pl* qui s'exécute en tout début d'installation.

os: contient la copie du répertoire I386 du CD d'installation de Windows XP, préparé selon les recommandations évoquées précédemment, c'est à dire à jour des hotfixes et des drivers..

scripts: contient les fichiers .bat pour déployer les applications additionnelles avec leur « silent installer »

packages: contient les exécutable des applications à installer (Office, Acrobat Reader, Firefox, ...)

updates : contient les hotfix récupérés par le script « *prepare* »

4.1.2 Noyau de démarrage

La distribution unattended de Sourceforge propose une image iso à graver sur CD permettant de faire démarrer le PC à installer depuis un cdrom, mais également un noyau de démarrage Linux que l'on peut mettre sur un serveur tftp pour permettre un démarrage en réseau via PXE. Le démarrage en réseau procure un grand confort d'utilisation (notamment ne pas avoir à enlever le cdrom du lecteur, lors du second reboot du PC). Des serveurs dhcp et tftp fournissent le noyau Linux et l'environnement unattended.

Nous avons dû utiliser le noyau de la version de développement CVS pour avoir un noyau 2.6.21 qui supporte le démarrage sur des disques SATA.

4.1.3 Peuplement du serveur Unattended

Pour « peupler » le serveur unattended, on lance un script appelé « *prepare FRA* » (argument FRA pour la paramétrisation française), qui se charge de récupérer par réseau et d'installer :

- une partie des logiciels additionnels qui seront installés lors de la post-installation (installés dans `~/install/packages/`),
- l'ensemble des mises à jour (update et hotfixes) pour les OS à installer dans `~/updates` (ceux ci ne nous servent pas car nous les avons intégré par

slipstreaming au préalable, mais la distribution propose de les installer en post installation).

On peut donc personnaliser la postinstallation en rajoutant des scripts et des exécutable dans les répertoires « *scripts* » et « *packages* ».

4.2 Coté client : démarrage en réseau du PC à installer

Lorsque le serveur unattended est prêt et configuré pour exporter le master Windows en CIFS, on peut faire démarrer le PC client sur le noyau linux fourni par la distribution unattended de sourceforge.

Le PC client démarre en PXE sur le noyau Linux de la distribution unattended. Après la séquence PXE, DHCP, TFTP traditionnelle, ce noyau Linux spécialement conçu a pour fonction de monter le partage CIFS `\\ntinstall\install` du serveur « unattended » sur un point de montage Z:, puis d'exécuter le script Perl `Z:\dosbin\install.pl`.

Ce script « *install.pl* » lancé avant le début de l'installation permet à l'administrateur de contrôler et préparer, certaines actions importantes dans le déroulement de l'installation. Il permet de :

- partitionner le disque comme on le souhaite (avec `gnu parted`),

- poser un ensemble de questions (clé de licence windows, nom du PC, mot de passe administrateur, jointure au domaine etc...), dont les réponses permettent de remplir les différentes sections du fichier d'installation *unattend.txt*,

- parcourir le répertoire `Z:\scripts` et d'afficher un menu permettant de choisir les applications additionnelles qui seront installées lors de la postinstallation,

- générer les 3 fichiers *unattend.txt*, *postinst.bat*, et *doit.bat* en charge de l'automatisation complète de l'installation.

- lancer une émulation de DOS (dosemu) permettant d'exécuter la commande `winnt /s:Z:\os\winxpsp2\i386 /u:C:\netinst\unattend.txt` qui démarre l'installation de windows XP en mode « unattended ». Les fichiers étant contenus dans le répertoire `\\ntinstall\install\os\ winxpsp2\I386\` mappé sur Z: sont alors copiés localement sur le disque de la machine à installer dans C:.

Une fois la copie terminée des binaires du système de Z: vers C:, la machine redémarre et le processus d'installation classique, mais « sans surveillance », de Windows se lance.

4.3 Particularité pour l'intégration des pilotes matériels avec le système Unattended de SourceForge

Comme indiqué plus haut, nous avons utilisé le programme `DPS_base.exe` fourni par le site DriverPack [Réf. 5]. Par défaut la méthode 2 décrite plus haut (pyron) n'est pas

compatible avec le système « unattended de SourceForge », car les drivers doivent être compressés dans un répertoire qui doit être à la racine du CD d'installation C:. Or, la distribution « unattended » travaille uniquement en dessous du répertoire I386 ce qui rend le répertoire [C:\OEM](#) inaccessible.

Aussi, pour rendre compatible la distribution « unattended » avec la méthode Pyron, nous avons déplacé le répertoire OEM généré par l'intégrateur de DriverPacks, dans le répertoire I386 du système de fichier Unattended.

Pour que la méthode Pyron fonctionne, il nous faut donc par la suite copier le répertoire I386\OEM vers C:. Pour cela nous avons rajouté un XCOPY dans le fichier « unattend.txt », qui copiera le répertoire I386\OEM/ et son contenu, à la racine du disque C:.

```
doit_cmds = "Z::cd Z:\os\winxpsp2\i386;xcopy OEM
c:\OEM /Y /E /I;
winnt /rx:lang /s:Z:\os\winxpsp2\i386 /u:C:\netinst\unattend
.txt"
```

5 Méthode d'installation unattended sur cdrom

Contrairement à la distribution Unattended de SourceForge, il n'existe pas de possibilité d'interagir avec un programme initial en début d'installation. Nous installons donc le même ensemble de logiciels additionnels prédéfini pour tous les PCs. Ce qui implique un choix commun d'applications occupant un espace disque réduit tenant sur un CD de 700Mo.

Nous avons dû prendre en compte les contraintes suivantes:

- compatibilité avec la configuration matérielle d'un maximum de PC (donc usage d'un cdrom et non d'un dvd, même si la méthode est identique),
- une seule interaction avec l'administrateur lors de l'installation. Le partitionnement est la seule phase où l'administrateur intervient manuellement.

Pour remplir ces objectifs il est nécessaire d'optimiser l'espace sur le cdrom.

L'ordre des phases pour préparer le master Windows est alors le suivant :

1. Intégration d'Internet Explorer 7 avec nLite
2. Intégration de Windows Média Player 11 avec WMP11 Integrator [Réf. 4]
3. Slipstreaming des hotfixes (scripts de commandes ou nLite)
4. Optimisation, allègement du master XP, et personnalisations avec nLite
5. Intégration d'un ensemble minimum strictment nécessaire de drivers (issus du site DriverPacks) : pack des drivers « disque » et « réseau »
6. Personnalisation du fichier *winnt.sif* et des fichiers dans les répertoires \$OEM\$ et \$\$
7. Intégration de ProInstall [Réf. 8] pour la post-installation de logiciels additionnels (on peut

choisir des produits « concurrents » comme Xplode ou WPI.

8. gravure du master Windows sur CDROM

Si les points 1 à 6 sont sensiblement communs à l'installation réseau unattended de SourceForge, en revanche, les points 7 et 8 sont spécifiques à la préparation du « master » sur CD.

5.1 La gravure du cdrom.

La création du cdrom bootable se fait grâce à l'outil freeware CDBurnerXP Pro [Réf. 9]. Il faut également récupérer les secteurs de boot. Différentes méthodes existent, mais la plus simple consiste à récupérer le dit fichier sur l'excellent site de JC Bellamy [Réf. 10]. Ensuite, dans CDBurner, il faut créer un nouveau disque de données, y glisser l'ensemble du contenu du répertoire préparé, et demander à ce que le cdrom produit soit bootable.

Les paramètres suivants seront nécessaires :

- pas d'émulation disque
- Load Segment à 0 (0x7C0 sur les autres logiciels de gravure)
- 4 Secteurs
- Valider : « Disable ISO File Delimiter » et « Enforce ISO Level 1 »

Il ne reste plus qu'à graver le CD et tester sur VMWare, ou un vrai PC.

6 Etape de Post-Installation : ajout de logiciels additionnels

Cette phase dépend en grande partie du système et du media d'installation qui est utilisé.

6.1 Les « silent installers »

L'installation automatique des logiciels additionnels repose sur la notion des « *silent installers* ». C'est à dire des arguments que l'on passe au logiciel pour effectuer une installation par défaut sans aucune interaction avec l'administrateur. Pour firefox par exemple ce sont les arguments *-ms -ira* qui provoquent une installation silencieuse. On trouvera sur de nombreux sites web [Réf. 7] une liste des principaux « argument silencieux » pour différents logiciels.

6.2 Pour ce qui est du système Unattended de SourceForge

L'intérêt de ce système est de pouvoir lancer et interagir avec un programme Perl avant le début de l'installation. Lors du premier démarrage du PC client à installer, un script perl *install.pl* parcourt le répertoire *~/install/script* à la recherche des scripts d'installation **.bat* et affiche un menu permettant de choisir les logiciels que l'on souhaite installer. Cette liste de logiciels est alors inscrite dans un

fichier « *todo.txt* » et traitée en fin d'installation par le script *postinst.bat*.

La postinstallation est réalisée grâce à la section [*GuiRunOnce*] du fichier *unattend.txt* qui permet de lancer la commande « *postinst.bat* » à la fin de l'installation du système Windows.

```
[GuiRunOnce]
Command0="C:\netinst\mapznrn.bat
C:\netinst\postinst.bat"
```

Le script *postinst.bat* est lancé et exécute un script Perl *todo.pl* qui va lancer la série d'installations contenues dans le fichier *C:\netinst\todo.txt*.

Pour lancer l'installation des logiciels en phase finale, il suffit en définitive d'alimenter deux répertoires :

- *~/install/packages* qui contient les binaires d'installation des applications souhaitées,
- *~/install/script* contenant les scripts qui seront exécutés en post-installation

Exemple de script d'installation *firefox.bat*

```
todo.pl "%Z%\packages\mozilla\firefox-2006-%WINLANG
%.exe -ms -ira"
```

6.3 Pour la méthode de préparation du master Windows sur CD

Nous utilisons l'outil ProInstall [Réf. 8]. Celui ci comporte deux parties: le « constructeur » (qui va produire le fichier *options.pro*) et « l'interpréteur » *proinstall.exe*

- **le constructeur** : l'outil permet graphiquement de générer le fichier « *options.pro* » qui contient les références des programmes dont il faut enchaîner l'installation.

Compte tenu de la place limitée disponible sur CD, nous avons réalisé la post-installation des produits suivants:

- hotfixes non-slipstreamable (KB890830, ...),
- la personnalisation de la base de registre (HKLM) avec des fichiers *.reg* préparés selon nos besoins,
- la recopie du répertoire I386 depuis le cdrom vers le disque dur,
- l'intégration de WindowsUpdate, des Frameworks .Net et de WindowsScript,
- l'installation de produits utiles communs freeware (Codecs, Firefox, Thunderbird, Spybot, SIW, IZArc et FoxitReader (AcrobatReader est trop volumineux pour le cdrom).

- **l'interpréteur proinstall** : Il suffit de copier le fichier « *options.pro* » généré via le constructeur dans le même répertoire (« *\$OEM\$\install* ») que l'interpréteur « *proinstall.exe* » et d'ajouter dans le fichier *winnt.sif* le lancement de *proinstall* dans la section *GuiRunOnce*

```
[GuiRunOnce]
```

```
ProInstall = "%systemdrive%\install\proinstall.exe"
```

7 Conclusions : avantages et inconvénients

7.1 Avantages de la distribution Unattended de SourceForge

Avec la distribution « unattended » de SourceForge, les objectifs sont atteints : un PC Windows XP est installé en une heure environ avec les logiciels additionnels... et « 0 clic ! ». On a un gain de temps énorme à l'installation.

- La méthode de préparation du master Windows telle que nous l'avons effectuée avec la méthode Pyron permet d'incorporer un grand nombre de pilotes, et prend en compte la diversité des matériels de PC différents avec une seule source d'installation (pas de gestion d'images disques comme avec les systèmes de clonage)

- Nous n'avons pas de limitation du volume du « Master Windows » sur le serveur puisqu'il réside sur un partage cifs sur disque, comme cela est le cas avec une installation par cdrom

- Le démarrage du PC client sur un noyau Linux permet une interaction avec l'administrateur grâce à un script Perl après le démarrage du PC à installer : on a alors le choix du partitionnement du disque et le choix des logiciels à installer dans des menus à l'écran.

- On a la possibilité de gérer des classes d'installation de PC (administratif, développement, scientifique...).

Malgré un travail initial préparatoire de paramétrage et de personnalisation du master Windows, le gain de temps est immédiatement rentabilisé lors des premières installations.

7.2 Inconvénients de la distribution Unattended de SourceForge

La mise au point de l'ensemble de la chaîne d'installation nécessite un travail d'ingénierie assez important pour prendre en compte les problèmes de mises à jour des « hotfix » et d'installation des pilotes.

- Le projet Unattended de SourceForge est un projet opensource, dont le suivi est parfois erratique au cours du temps, avec une documentation assez dispersée

- La version stable actuelle 4.6 ne supporte pas le SATA, il faut prendre la version de développement 4.7 avec un noyau Linux 2.6.21 pour avoir le support SATA au démarrage. Il peut arriver que la version en cours de développement proposée sur le serveur CVS ne soit pas toujours stable, cependant le projet évolue et les erreurs sont corrigées.

7.3 Inconvénients du cdrom

- La limitation de l'espace disque disponible sur CD implique un effort supplémentaire de sélection et de tests des drivers et des applications à installer

- Nous sommes parfois tombé sur des incompatibilités entre les différents logiciels que nous mettons sur le CD. Ces incompatibilités évoluent avec les versions de logiciels et leur ordre d'insertion sur le CD

- Dans le cas d'un CD unattended on a une classe d'installation unique de PC. On ne peut pas gérer actuellement plusieurs classes d'installations différentes. On pourrait contourner cette limitation par une seconde phase d'interaction avec l'administrateur à la fin de l'installation pour sélectionner un profil de poste souhaité parmi plusieurs (un dvd serait alors requis).

7.4 Unattended vs clônage ?

La méthode d'installation sans surveillance fait gagner du temps pour l'installation ou la ré-installation fréquente de PC mais elle reste une technique d'« installation ». Elle n'est pas un déploiement d'images binaire de disques. Elle prend donc plus de temps pour être déployée sur disque et rendre le PC opérationnel.

Cette technique sans surveillance ne peut pas être envisagée pour le déploiement massif et parallèle de plusieurs dizaines de PC (salles d'enseignement) comme le permettent des systèmes de clônage en multicast tel que Rembo.

En revanche le fait de travailler sur un master Windows unique qui contient un ensemble très large de pilotes matériels, permet de ne pas avoir à gérer de nombreuses images disques. C'est un atout pour des entités qui possèdent un parc de PC très hétérogène plutôt que de nombreuses salles de PC identiques.

Dans les deux cas on aboutit à des PC installés à l'identique ce qui est d'une réelle efficacité dans la gestion d'un parc hétérogène, et à une industrialisation du procédé d'installation qui procure un parc homogène et une maintenance ultérieure facilitée.

Bibliographie

- [1] Site pour nLite <http://www.nliteos.com/>
- [2] Méthode d'installation à T-12 avec cmdlines.txt <http://www.winattended.com/hotfixswindxp.php>
- [3] Séquence temporelle des phases d'installation de Windows <http://unattended.free.fr/timeline.htm>
- [4] Site pour WUD <http://wud.jcarle.com/>
- [5] Site de récupération des pilotes matériels <http://www.driverpacks.net/DriverPacks/>
- [6] Méthode d'installation des drivers de Pyron <http://www.msfn.org/board/index.php?showtopic=12566>
- [7] Site pour les « silent installers » http://wpkg.org/index.php/Category:Silent_Installers
- [8] Site pour ProInstall (téléchargement/ contributions) <http://www.win-web.be>
- [9] Site pour CDBurnerXP <http://www.cdburnerxp.se/>
- [10] site Bellamy : cd bootable <http://jc.bellamy.free.fr/fr/cdbootable.html>