

# Gestion des applications Windows par MSI

Introduction à l'approche pratiquée à  
l'INP de Toulouse

Capitoul – 12/10/2006

Giles Carré

# Objectifs initiaux

- Pouvoir déployer, mettre à jour et supprimer les applications selon les besoins
- Mutualiser au niveau de l'INP
- Utiliser le même packaging MSI pour tout type de déploiement (auto/manu, autonome/AD, tout chemin, etc)
- Non seulement les applications : tout fichier, configuration, y/c développé par nous

# L'offre MSI (1/2)

## ➤ La solution idéale

- L'éditeur livre son application sous forme de MSI
- Accompagné d'un outil d'adaptation (pré-configuration, réglages, etc)
- Exemples :
  - IEAK pour Microsoft IE
  - ORK pour Microsoft Office
  - VirusScan (McAfee/NAI)
  - Adobe Reader

# L'offre MSI (2/2)

## ➤ La solution intermédiaire

- L'éditeur livre son application sous forme de MSI
- Il va falloir l'adapter soi-même (installation silencieuse, réglages)

## ➤ La solution lourde

- Il faut se fabriquer soi-même le packaging (« repackaging » d'application)

# Les deux approches MSI (1/3)

## ➤ MSI éditeur

- Généré avec des bibliothèques spécialisées
- L'éditeur **sait** ce qu'il veut obtenir ; c'est une méthode « propre »

# Les deux approches MSI (2/3)

## ➤ Repackaging

- C'est l'administrateur qui s'y colle
- Il va tenter de comprendre ce que fait le processus d'installation fourni par l'éditeur et d'en déduire un déroulement d'installation

# Les deux approches MSI (3/3)

- Les deux méthodes sont totalement différentes
  - Même si la structure de l'objet obtenu (le MSI) est la même
  - La qualité est très différente

# Repackaging

- Le processus d'installation d'application est variable selon les instances d'exécution (configuration du poste, etc)
- Il va donc falloir capter les actions avec le risque
  - De capter trop d'actions spécifiques
  - De ne pas capter des actions qui seraient nécessaires sur d'autres postes
- Puis nettoyer ou compléter
- ➔ **INDISPENSABLE** : un outil de haut niveau



# Les outils (1/2)

- Livré en standard : WinInstall LE
  - Suffisant pour un repackaging exceptionnel
  - Pas assez d'outils d'aide à la mise au point
  - Trop de limitations telles que la taille de l'application à traiter (incapable de créer un paquetage pour Matlab, qui contient plusieurs dizaine de milliers de fichiers)

# Les outils (2/2)

- Notre choix : Wise Package Studio
  - Proche de Microsoft
  - Puissant
  - Version d'évaluation complète téléchargeable et valable un mois
  - Cher

# Notre démarche (1/2)

- Réaliser plusieurs dizaines de paquetages en à peu près 3 mois
- Une démarche en plusieurs phases
  - Phase 1 : initialiser le processus (juin 2005)
    - Recherche d'une compétence (SSII) maîtrisant la technologie et achat d'une prestation (Cap Laser à Castres)
      - De fourniture de 5 paquetages
      - Réalisés en nos murs en collaboration avec les informaticiens de l'INP
      - Transfert de compétence par l'exemple
      - → prestation très efficace

# Notre démarche (2/2)

- Phase 2 (juillet à septembre 2005)
  - Réalisation de la plupart des paquetages (une quarantaine)
  - À deux personnes (Béatrice Guinard, Guillaume Podevin)
- Formation d'une semaine chez Altiris (nov 2005)
- Puis reprise des paquetages :
  - En cours d'année pour correctifs sur quelques'uns
  - Tout les autres pour la rentrée 2006

# Acteurs de la réalisation des MSI

- Idée initiale : tous ceux qui ont besoin de réaliser un déploiement d'application, notamment pour gérer la réactivité
- Après expérimentation :
  - Pas réaliste (technicité requise)
  - Donc, à l'INP, dualité des compétences :
    - Un technicien « transversal », formé chez Wise pour le support, le conseil et la connaissance du produit
    - Un technicien local, connaissant très bien les produits à installer (notamment toutes les mises au point à effectuer lors d'une installation manuelle)

# Déroulement de la démarche

- Architecture choisie : modularité maximale
  - Un RIS minimal
  - Tout le reste par
    - GPO (réglages, déploiement MSI)
    - Objets sur serveurs de fichiers (scripts, partie d'applications, etc)
- Cette démarche ne souffre pas d'écart
  - Que faire si on n'arrive pas à réaliser un paquetage (ou qu'on en a pas le temps)

# Autre déroulement possible

- Conserver un système de déploiement existant (clonage, RIS « dopé », etc)
- Convertir progressivement les applications en MSI
- Mais ...
  - On risque de n'aller jamais au bout
  - Transition à gérer (être capable de désinstaller une application installée par RIS et remplacée par une installation par MSI)

# Méthodes de déploiement

- **Simple GPO dans Active Directory**
  - Facile à mettre en œuvre
  - Remontée d'information difficile
- **Microsoft SMS**
  - Apport de la gestion de version
  - Retour de l'état des déploiements
- **Outils propriétaires**
  - À éviter, car chers et pas obligatoirement pérennes



# Problèmes de déploiement rencontrés

- Application aléatoire des GPO (problèmes réseau)
- Mutualisation
  - Problème inter-site
- Gestion des versions/révisions
  - Plusieurs versions/révisions d'une même application simultanément déployées (licence, contraintes techniques, etc)
  - → nommage des paquetages
  - → gestion des révisions

# Maintenance des applications

- Objectif initial : pouvoir
  - Réinstaller/modifier une application
  - Modifier une configuration
  - Avec une grande réactivité
- Objectif non encore atteint
  - On arrive tout juste au bout du repackaging initial
  - Difficulté technique : désinstallation (mais ce problème semble être maintenant résolu)

# Modularité

- Objectif : un seul MSI pour plusieurs déclinaisons d'une même application
- Exemple : Matlab et ses Toolboxes
  - droit d'utilisation des toolboxes différent selon l'utilisateur ou la machine
  - 2 approches :
    - Plusieurs MSI selon les toolboxes à déployer → trop lourd
    - Un seul MSI avec toutes les toolboxes, puis un MST pour modifier le droit d'utilisation (licence)

# Et si c'était à refaire ?

- Nous n'avons pas encore de visibilité formelle sur le retour sur investissement
- Mais ...
  - Nous apprécions déjà
    - L'automatisation
    - La réutilisabilité (autre type de cible que celles prévues lors de la fabrication du MSI)
  - C'est un investissement pérenne
  - Dans la démarche, on aurait aussi pu attendre qu'il y ait plus de MSI éditeur

# Et puis après...

- Pourquoi ne pas mutualiser les paquetages MSI en inter-universitaire ?
- Pourquoi acheter, déployer des applications complexes ou à usage rare et qui pourraient déjà exister ailleurs (les ateliers inter-U, les autres universités, le CICT) ? → notre rôle d'administrateur == sensibiliser les responsables de formation ou de scolarité, les chercheurs

# Gestion des applications Windows par MSI

Introduction à l'approche pratiquée à  
l'INP de Toulouse  
Capitoul – 12/10/2006  
Giles Carré

# Objectifs initiaux

- Pouvoir déployer, mettre à jour et supprimer les applications selon les besoins
- Mutualiser au niveau de l'INP
- Utiliser le même packaging MSI pour tout type de déploiement (auto/manu, autonome/AD, tout chemin, etc)
- Non seulement les applications : tout fichier, configuration, y/c développé par nous

# L'offre MSI (1/2)

## ➤ La solution idéale

- L'éditeur livre son application sous forme de MSI
- Accompagné d'un outil d'adaptation (pré-configuration, réglages, etc)
- Exemples :
  - IEAK pour Microsoft IE
  - ORK pour Microsoft Office
  - VirusScan (McAfee/NAI)
  - Adobe Reader



## L'offre MSI (2/2)

- La solution intermédiaire
  - L'éditeur livre son application sous forme de MSI
  - Il va falloir l'adapter soi-même (installation silencieuse, réglages)
- La solution lourde
  - Il faut se fabriquer soi-même le packaging (« repackaging » d'application)

# Les deux approches MSI (1/3)

## ➤ MSI éditeur

- Généré avec des bibliothèques spécialisées
- L'éditeur **sait** ce qu'il veut obtenir ; c'est une méthode « propre »

# Les deux approches MSI (2/3)

## ➤ Repackaging

- C'est l'administrateur qui s'y colle
- Il va tenter de comprendre ce que fait le processus d'installation fourni par l'éditeur et d'en déduire un déroulement d'installation

## Les deux approches MSI (3/3)

- Les deux méthodes sont totalement différentes
  - Même si la structure de l'objet obtenu (le MSI) est la même
  - La qualité est très différente

# Repackaging

- Le processus d'installation d'application est variable selon les instances d'exécution (configuration du poste, etc)
- Il va donc falloir capter les actions avec le risque
  - De capter trop d'actions spécifiques
  - De ne pas capter des actions qui seraient nécessaires sur d'autres postes
- Puis nettoyer ou compléter
- ➔ **INDISPENSABLE** : un outil de haut niveau

# Les outils (1/2)

- Livré en standard : WinInstall LE
  - Suffisant pour un repackaging exceptionnel
  - Pas assez d'outils d'aide à la mise au point
  - Trop de limitations telles que la taille de l'application à traiter (incapable de créer un paquetage pour Matlab, qui contient plusieurs dizaine de milliers de fichiers)

## Les outils (2/2)

- Notre choix : Wise Package Studio
  - Proche de Microsoft
  - Puissant
  - Version d'évaluation complète téléchargeable et valable un mois
  - Cher

# Notre démarche (1/2)

- Réaliser plusieurs dizaines de paquetages en à peu près 3 mois
- Une démarche en plusieurs phases
  - Phase 1 : initialiser le processus (juin 2005)
    - Recherche d'une compétence (SSII) maîtrisant la technologie et achat d'une prestation (Cap Laser à Castres)
      - De fourniture de 5 paquetages
      - Réalisés en nos murs en collaboration avec les informaticiens de l'INP
      - Transfert de compétence par l'exemple
      - → prestation très efficace



## Notre démarche (2/2)

- Phase 2 (juillet à septembre 2005)
  - Réalisation de la plupart des paquetages (une quarantaine)
  - À deux personnes (Béatrice Guinard, Guillaume Podevin)
- Formation d'une semaine chez Altiris (nov 2005)
- Puis reprise des paquetages :
  - En cours d'année pour correctifs sur quelques'uns
  - Tout les autres pour la rentrée 2006

# Acteurs de la réalisation des MSI

- Idée initiale : tous ceux qui ont besoin de réaliser un déploiement d'application, notamment pour gérer la réactivité
- Après expérimentation :
  - Pas réaliste (technicité requise)
  - Donc, à l'INP, dualité des compétences :
    - Un technicien « transversal », formé chez Wise pour le support, le conseil et la connaissance du produit
    - Un technicien local, connaissant très bien les produits à installer (notamment toutes les mises au point à effectuer lors d'une installation manuelle)

# Déroulement de la démarche

- Architecture choisie : modularité maximale
  - Un RIS minimal
  - Tout le reste par
    - GPO (réglages, déploiement MSI)
    - Objets sur serveurs de fichiers (scripts, partie d'applications, etc)
- Cette démarche ne souffre pas d'écart
  - Que faire si on n'arrive pas à réaliser un paquetage (ou qu'on en a pas le temps)

## Autre déroulement possible

- Conserver un système de déploiement existant (clonage, RIS « dopé », etc)
- Convertir progressivement les applications en MSI
- Mais ...
  - On risque de n'aller jamais au bout
  - Transition à gérer (être capable de désinstaller une application installée par RIS et remplacée par une installation par MSI)

# Méthodes de déploiement

- **Simple GPO dans Active Directory**
  - Facile à mettre en œuvre
  - Remontée d'information difficile
- **Microsoft SMS**
  - Apport de la gestion de version
  - Retour de l'état des déploiements
- **Outils propriétaires**
  - À éviter, car chers et pas obligatoirement pérennes

# Problèmes de déploiement rencontrés

- Application aléatoire des GPO (problèmes réseau)
- Mutualisation
  - Problème inter-site
- Gestion des versions/révisions
  - Plusieurs versions/révisions d'une même application simultanément déployées (licence, contraintes techniques, etc)
  - → nommage des paquetages
  - → gestion des révisions

# Maintenance des applications

- Objectif initial : pouvoir
  - Réinstaller/modifier une application
  - Modifier une configuration
  - Avec une grande réactivité
- Objectif non encore atteint
  - On arrive tout juste au bout du repackaging initial
  - Difficulté technique : désinstallation (mais ce problème semble être maintenant résolu)

# Modularité

- Objectif : un seul MSI pour plusieurs déclinaisons d'une même application
- Exemple : Matlab et ses Toolboxes
  - droit d'utilisation des toolboxes différent selon l'utilisateur ou la machine
  - 2 approches :
    - Plusieurs MSI selon les toolboxes à déployer → trop lourd
    - Un seul MSI avec toutes les toolboxes, puis un MST pour modifier le droit d'utilisation (licence)



# Et si c'était à refaire ?

- Nous n'avons pas encore de visibilité formelle sur le retour sur investissement
- Mais ...
  - Nous apprécions déjà
    - L'automatisation
    - La réutilisabilité (autre type de cible que celles prévues lors de la fabrication du MSI)
  - C'est un investissement pérenne
  - Dans la démarche, on aurait aussi pu attendre qu'il y ait plus de MSI éditeur

## Et puis après...

- Pourquoi ne pas mutualiser les paquetages MSI en inter-universitaire ?
- Pourquoi acheter, déployer des applications complexes ou à usage rare et qui pourraient déjà exister ailleurs (les ateliers inter-U, les autres universités, le CICT) ? → notre rôle d'administrateur == sensibiliser les responsables de formation ou de scolarité, les chercheurs