Politique de test – Network Pro

|  |  |
| --- | --- |
| **Titre du document** | Politique\_test\_np.docx |
| **Auteur du document** | Ivan KLARMAN |
| **Validation** | Frédéric SANANES |
| **Version du document** | V1.0 |

# Objectifs du document

La politique de test vise à appréhender les différents dysfonctionnements du logiciel avant, pendant et après la livraison du produit logiciel.  
Il permet de concevoir la robustesse du logiciel face aux différents bugs qu’ils pourraient exister.

# Contraintes technique du logiciel

Les contraintes actuelles sont les suivantes :

* **Environnement des postes clients**
  + **Windows 7 Pro x64 et x86**
* **Environnement des serveurs**
  + **Windows server 2008 R2**
  + **Windows server 2012 R2**
* **Applications à déployer**
  + **Liste non défini** à l’heure actuelle, des applications à déployer par le client
* **Déploiement du logiciel**
  + Le logiciel sera installé sur **un serveur par domaine réseau** (1 pour l’administratif, 1 pour le pédagogique) et par **site** (4 sites au total).
* **Droit d’accès**
  + L’accès au logiciel doit utiliser les droits provenant de l’annuaire **Active Directory**.

# Tests fonctionnels

Le logiciel doit respecter les exigences du client et réaliser les fonctionnalités demandées.  
Pour cela un cahier de recette a été rédigé « **cahier\_recette\_np.xlsx »**

# Tests unitaires

## Partie « Etat du réseau »

Le logiciel doit effectuer un scan du réseau dans le domaine réseau sur lequel il va être déployé.

Voici une liste de tests unitaire à réaliser :

* Créer une fonction qui liste l’ensemble des machines d’un domaine réseau.

Aucune entrée.  
**Exemple de sortie attendue** : List<Machine> // liste d’objets Machine

* Créer une fonction qui test si la machine est un serveur ou un poste client. On aura en paramètre une adresse IP correspondant à une chaîne de caractères.  
  **Exemple d’entrée :** 192.168.54.3 // Chaîne de caractères  
  **Exemple de sortie attendue** : Serveur // Chaîne de caractères  
  **Exemple de sortie attendue** : Client // Chaîne de caractères  
  **Exemple de sortie alternatif** : Autre // Chaîne de caractères
* Créer une fonction qui fournit la date et heure du dernier redémarrage de la machine.

**Exemple d’entrée :** 192.168.54.3 // Chaîne de caractères  
**Exemple de sortie attendue** : 27/12/2015 17:25:00 // Date (Version française)

* Créer une fonction permettant de récupérer à partir d’une adresse IP de connaître le nom de la machine, liste des utilisateurs et le dernier redémarrage du serveur.  
  **Exemple d’entrée :** 192.168.54.3 // Chaîne de caractères  
  **Exemple de sortie attendue** : oMachine // Objet de type Machine avec les valeurs.

On vérifiera que les variables Name, List<User> et LastRestart contiennent bien les valeurs attendus.  
Name // Chaîne de caractères  
List<User> // Liste des utilisateurs

LastRestart // Date

* Créer une fonction qui récupère l’état d’une machine.  
  Trois valeurs sont possibles :   
  - Démarré  
  - Non disponible  
  - Eteint

**Exemple d’entrée :** 192.168.54.3 // Chaîne de caractères  
**Exemple de sortie attendue** : Démarré // Chaîne de caractères  
**Exemple de sortie attendue** : Non disponible // Chaîne de caractères

## Partie « Wake On Lan »

Le logiciel doit pouvoir planifier l’allumage, extinction et le redémarrage d’un poste client ou serveur.

Voici une liste de tests unitaire à réaliser :

* Créer une fonction qui éteint une machine.  
  Ne seront acceptés que les machines allumées.  
  **Exemple d’entrée :** 192.168.54.3 // Chaîne de caractères  
  **Exemple de sortie attendue** : En cours d’extinction // Chaîne de caractères  
  **Exemple de sortie alternatif** : Impossible, la machine est déjà éteinte. // Chaîne de caractères
* Créer une fonction qui allume une machine.  
  Ne seront acceptés que les machines éteintes.  
  **Exemple d’entrée :** 192.168.54.3 // Chaîne de caractères  
  **Exemple de sortie attendue** : En cours de démarrage // Chaîne de caractères  
  **Exemple de sortie alternatif** : Impossible, la machine est déjà allumée. // Chaîne de caractères
* Créer une fonction qui redémarre une machine.  
  Ne seront acceptés que les machines démarrées.  
  **Exemple d’entrée :** 192.168.54.3 // Chaîne de caractères  
  **Exemple de sortie attendue** : Démarré // Chaîne de caractères  
  **Exemple de sortie alternatif** : Impossible, la machine est éteinte. // Chaîne de caractères  
    
  A noter, que l’on sort de la fonction quand la machine c’est éteinte, puis c’est rallumé.
* Créer une fonction qui crée une planification sur une machine.  
  Trois valeurs sont possibles pour le type de planification :  
  - Allumer  
  - Redémarrer  
  - Eteindre  
  **Exemple d’entrée :** 192.168.54.3, Allumer, 16/02/2016 17:15:12 // Adresse IP (Chaîne de caractères), Type de planification (Chaîne de caractères), Date.  
  **Exemple d’entrée :** 192.168.54.3, Redémarrer, 16/02/2016 17:15:12 // Adresse IP (Chaîne de caractères), Type de planification (Chaîne de caractères), Date.  
  **Exemple de sortie attendue** : oPlanif // Objet de type Planification  
  **Exemple de sortie alternatif** : Impossible, la machine à déjà une planification à ce jour et cette heure-là. // Chaîne de caractères  
    
  On prévoira une marge de **10 minutes** entre chaque planification pour une même machine.
* Créer une fonction qui liste l’ensemble des planifications sur l’ensemble des machines.  
  Aucune entrée.  
  **Exemple de sortie attendue** : List<Planification> // Liste d’objet de type Planification
* Créer une fonction qui annule une planification.  
  **Exemple d’entrée :** oPlanif // Objet de type Planification  
  **Exemple de sortie attendue** : Annulé // Chaîne de caractères

## Partie « Bureau à distance »

* Créer une fonction qui se connecte en bureau à distance à une machine.  
  Ne seront acceptés que les machines allumées.  
  **Exemple d’entrée :** 192.168.54.3 // Chaîne de caractères  
  **Exemple de sortie attendue** : Bureau à distance lancé // Chaîne de caractères

## Partie « Déploiement d’applications »

Le logiciel prévoit de déployer des applications en .exe x64 bits et x32 bits ou .msi x64 bits ou x32 bits sur des machines.  
  
Voici une liste de tests unitaire à réaliser :

* Créer une fonction qui liste les applications installés sur un poste client.  
  Ne seront acceptés que les postes clients démarrés.  
  **Exemple d’entrées :** oMachine // Objet de type Machine  
  **Exemple de sortie attendue** : List<App> // Liste d’objets de type Application
* Créer une fonction qui récupère la liste des applications prêtes à être déployés.  
  **Exemple d’entrée :** C:\Users\Ivan\AppToDeploy // Chaîne de caractères – Chemin vers le dossier des applications à déployer.

**Exemple de sortie attendue** : List<App> // Listes des applications

* Créer une fonction qui récupère et initialise les informations d’une application.  
  Notamment le nom, l’auteur, la version, l’extension et la compatibilité de l’application.  
  Ne seront acceptés que les extensions .exe et .msi.  
  Ne seront acceptés que les compatibilités x64 bits et x32 bits.  
  **Exemple d’entrée :** oApp // Objet de type Application
* Créer une fonction qui permet de déployer une application sur un poste client.  
  Ne seront acceptés que les postes clients démarrés.  
  **Exemple d’entrées :** oMachine, oApp // Objet de type Machine et un objet de type Application  
  **Exemple de sortie attendue** : Réussi // Chaîne de caractères  
  **Exemple de sortie attendue** : Déjà installé // Chaîne de caractères  
  **Exemple de sortie alternatif** : Echec - <Erreur> // Chaîne de caractères et explication de l’erreur.
* Créer une fonction qui lance l’application et vérifie que le processus de l’application existe sur un poste client. **– Optionnel**  
  Ne seront acceptés que les postes clients démarrés.  
  **Exemple d’entrées :** oMachine // Objet de type Machine  
  **Exemple de sortie attendue** : En marche // Chaîne de caractères  
  **Exemple de sortie alternatif** : Echec - <Erreur> // Chaîne de caractères et explication de l’erreur.
* Créer une fonction qui permet de mettre à jour une application sur un poste client.  
  Ne seront acceptés que les postes clients démarrés. **- Optionnel**  
  Ne seront acceptés que les extensions .exe et .msi.  
  Ne seront acceptés que les compatibilités x64 bits et x32 bits.  
  **Exemple d’entrées :** oMachine, oAppExistant, oAppPatch // Objet de type Machine, Objet de type Application, Objet de type Application  
  **Exemple de sortie attendue** : Réussi // Chaîne de caractères  
  **Exemple de sortie attendue** : Déjà installé // Chaîne de caractères  
  **Exemple de sortie alternatif** : Echec - <Erreur> // Chaîne de caractères et explication de l’erreur.

## Partie « Planification de déploiement d’application »

* Créer une fonction qui créée une planification de déploiement d’une application sur une machine.  
  **Exemple d’entrée :** 192.168.54.3, oApp, 16/02/2016 17:15:12 // Adresse IP (Chaîne de caractères), Type de planification (Chaîne de caractères), Date.  
  **Exemple de sortie attendue** : oPlanif // Objet de type Planification  
  **Exemple de sortie alternatif** : Impossible, la machine à déjà une planification à ce jour et cette heure-là. // Chaîne de caractères  
    
  On prévoira une marge de **10 minutes** entre chaque planification pour une même machine.
* Créer une fonction qui liste l’ensemble des planifications d’applications sur l’ensemble des machines.  
  Aucune entrée.  
  **Exemple de sortie attendue** : List<Planification> // Liste d’objet de type Planification