Configuration Manager – Client Health Remediation Script.

|  |  |  |  |
| --- | --- | --- | --- |
| Suivi de l’état du document | | |  |
| **Version** | **Date** | **Rédacteur** | **Modification** |
| V 1.0 | 13/07/2023 | Marc-Antoine ROBIN | Création du document |

**SOMMAIRE**

[1. Introduction 3](#_Toc149385380)

[1.1 Objet du document 3](#_Toc149385381)

[1.2 Périmètre d’application du document 3](#_Toc149385382)

[1.3 Prérequis 3](#_Toc149385383)

[2. Fonctionnement 4](#_Toc149385384)

[2.1 Résumé 4](#_Toc149385385)

[2.2 Déploiement GPO 5](#_Toc149385386)

[2.3 Exécution du script 7](#_Toc149385387)

[2.4 Fichier de configuration 9](#_Toc149385388)

[1. LocalFiles 9](#_Toc149385389)

[2. Client 9](#_Toc149385390)

[3. ClientInstallProperty 11](#_Toc149385391)

[4. Log 12](#_Toc149385392)

[5. Option 13](#_Toc149385393)

[6. Service 18](#_Toc149385394)

[7. Remediation 19](#_Toc149385395)

[3. Rapport 21](#_Toc149385396)

[4. Extension de console 22](#_Toc149385397)

[5. Annexes 23](#_Toc149385398)

[5.1 Base de données 23](#_Toc149385399)

[5.2 Web Service 25](#_Toc149385400)

[1. Prérequis 25](#_Toc149385401)

[2. Installation 25](#_Toc149385402)

[3. Erreur 500 28](#_Toc149385403)

[5.3 Extension de console 29](#_Toc149385404)

# Introduction

## Objet du document

**Ce document a pour objectif de détailler le fonctionnement du script de remédiation du client Configuration Manager déployé par GPO.**

**Ce processus est mis en place, premièrement afin d’installer le client Configuration Manager sur les machines ayant échappé au déploiement du client (GPO/Autre).**

**Le second but est de réparer les éventuels clients non sains qui ne peuvent pas communiquer correctement avec l’infrastructure et ainsi améliorer les taux de déploiement.**

**Enfin, les machines des domaines hors production étant fabriquées depuis l’infrastructure de production, le script permet de relier le client Configuration Manager à l’infrastructure de hors production une fois la jonction au domaine effectuée.**

La solution est basée sur le [travail d’Anders Rodland](https://www.andersrodland.com/configmgr-client-health/) et a été modifiée pour les besoins de GROUP.

## Périmètre d’application du document

## Prérequis

# Fonctionnement

## Résumé

L’outil est déployé par GPO afin de toucher les postes de travail des domaines cibles.

Les fichiers sont installés dans le dossier **C:\ProgramData\ConfigMgrClientHealth** et une tâche planifiée est créée afin d’exécuter le script régulièrement.

Le script est lancé via la tâche planifiée, elle-même déclenchée lors d'une connexion au réseau d’entreprise (**VPN inclus**).

Chaque test et étape de remédiation peut être activé ou non en modifiant un fichier de configuration xml.

Le script installe le client et/ou règle autant de problèmes connus que possible avant d'envoyer un rapport à un Web Service (**https://<ManagementPoint>/ConfigMgrClientHealth**).

Le Web Service se charge ensuite d'enregistrer ces données dans une base de données (**ClientHealth**) sur le serveur SQL de l’infrastructure Configuration Manager en utilisant un compte de service.

Enfin, cette base de données sert à produire un rapport montrant le résultat des exécutions du script sur chaque machine (voir [Rapport](#_Rapport))

L’exécution du script peut être suivi grâce au log créé sur chaque machine dans le dossier suivant :

**C:\ProgramData\ConfigMgrClientHealth\Logs**

Une fois le client installé et les problèmes corrigés, la tâche planifiée est désactivée.

En dehors de l’exécution du script par GPO, le script peut être exécuté depuis une extension de la console Configuration Manager.

Voir les chapitres suivants pour plus d’informations.

## Déploiement GPO

Paramètres passés au script

|  |  |  |
| --- | --- | --- |
| Paramètre | Description | Valeur par défaut |
| TaskName | Nom de la tâche planifiée à inscrire | ConfigMgr Client Health Remediation Script |
| InstallFolder | Dossier où installer la solution | C:\ProgramData\ConfigMgrClientHealth |
| LogFolder | Dossier où écrire le log d’exécution | C:\ProgramData\ConfigMgrClientHealth\Logs |
| Force | Permet de forcer l’installation du script de remédiation sans vérifier l’état du client | False |

Les sources sont copiées localement en avance de phase en utilisant la ressource suivante :

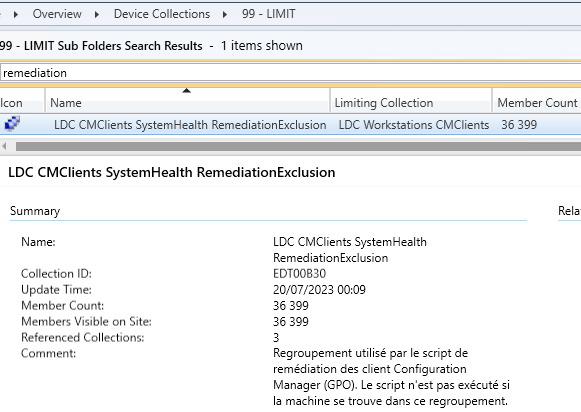
L’attribution d’un point de gestion Configuration Manager (MP) différent suivant le domaine de chaque machine permet de gérer la répartition des charges liées au Web Service mis en place pour la solution.

|  |  |  |  |
| --- | --- | --- | --- |
| Environnement | Domaine | MP | Fichier de configuration |
| Production |  | https://<MP>.fr | config-Client-Prod.xml |
| Production |  | https:// <MP>.fr | config-Client-Prod.xml |
| Production |  | https:// <MP>.fr | config-Client-Prod.xml |
| Production |  | https:// <MP>.fr | config-Client-Prod.xml |
| Production |  | https:// <MP>.fr | config-Client-Prod.xml |
| Hors production |  | https:// <MP>.fr | config-Client-HorsProd.xml |
| Hors production |  | https:// <MP>.fr | config-Client-HorsProd.xml |
| Hors production |  | https:// <MP>.fr | config-Client-HorsProd.xml |
| Hors production |  | https:// <MP>.fr | config-Client-HorsProd.xml |
| Hors production |  | https:// <MP>.fr | config-Client-HorsProd.xml |

Un test supplémentaire est effectué pour les postes de travail de production afin d’éviter de déployer la solution sur des clients sains.

Seules les postes de travail ne faisant pas partie du regroupement « **LDC CMClients SystemHealth RemediationExclusion**» pourrons installer la tâche planifiée.

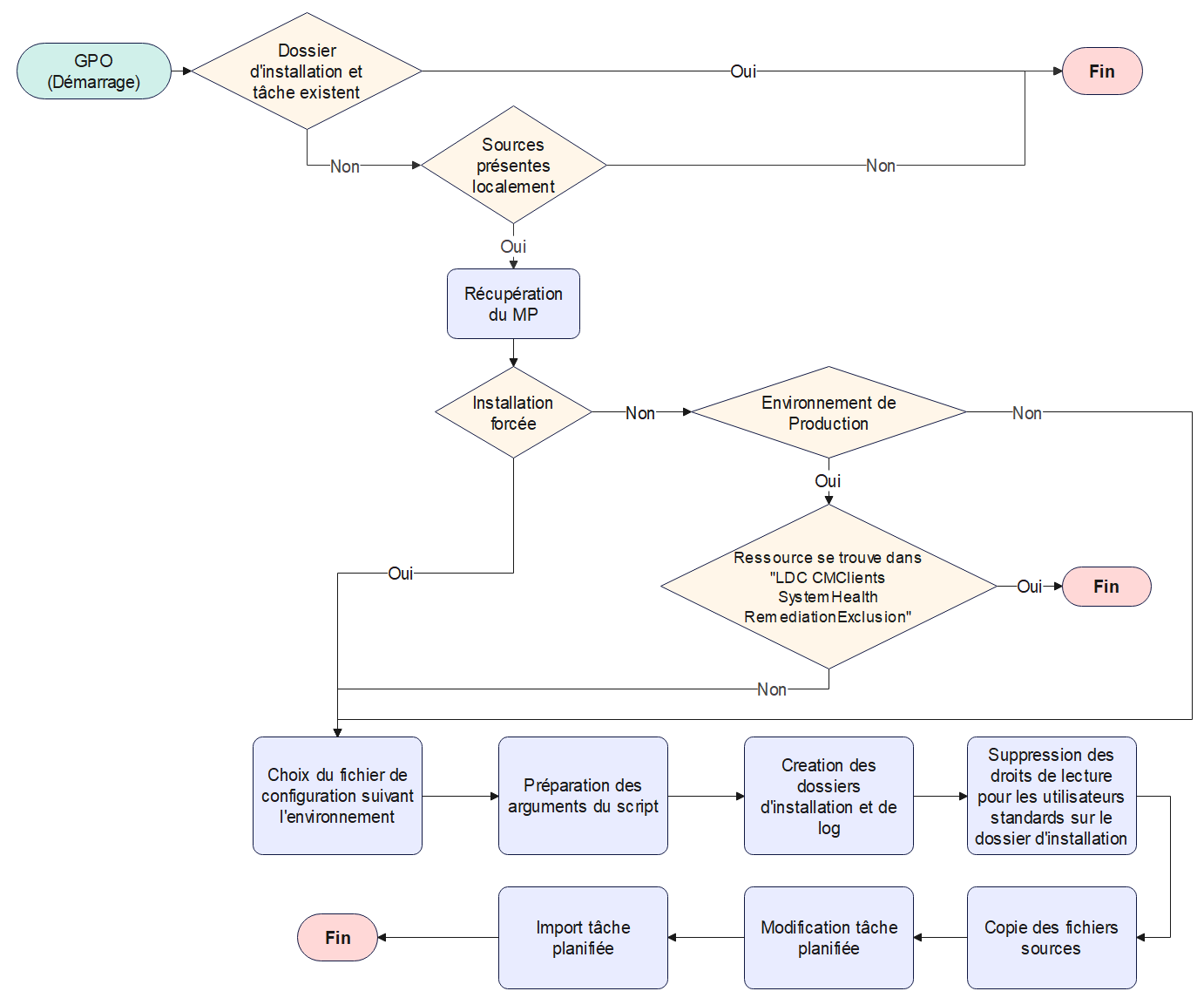
Les ressources ne faisant pas partie de ce regroupement :

* N’ont pas de client Configuration Manager
* Ont un client Configuration Manager non sain
* Ne se trouvent pas dans la version 22H2 de Windows 10

Les droits de lecture sont bloqués pour les utilisateurs standards sur les dossiers contenant les scripts :

* C:\ProgramData\ConfigMgrClientHealth
* C:\ProgramData\ConfigMgrClientHealth\Logs

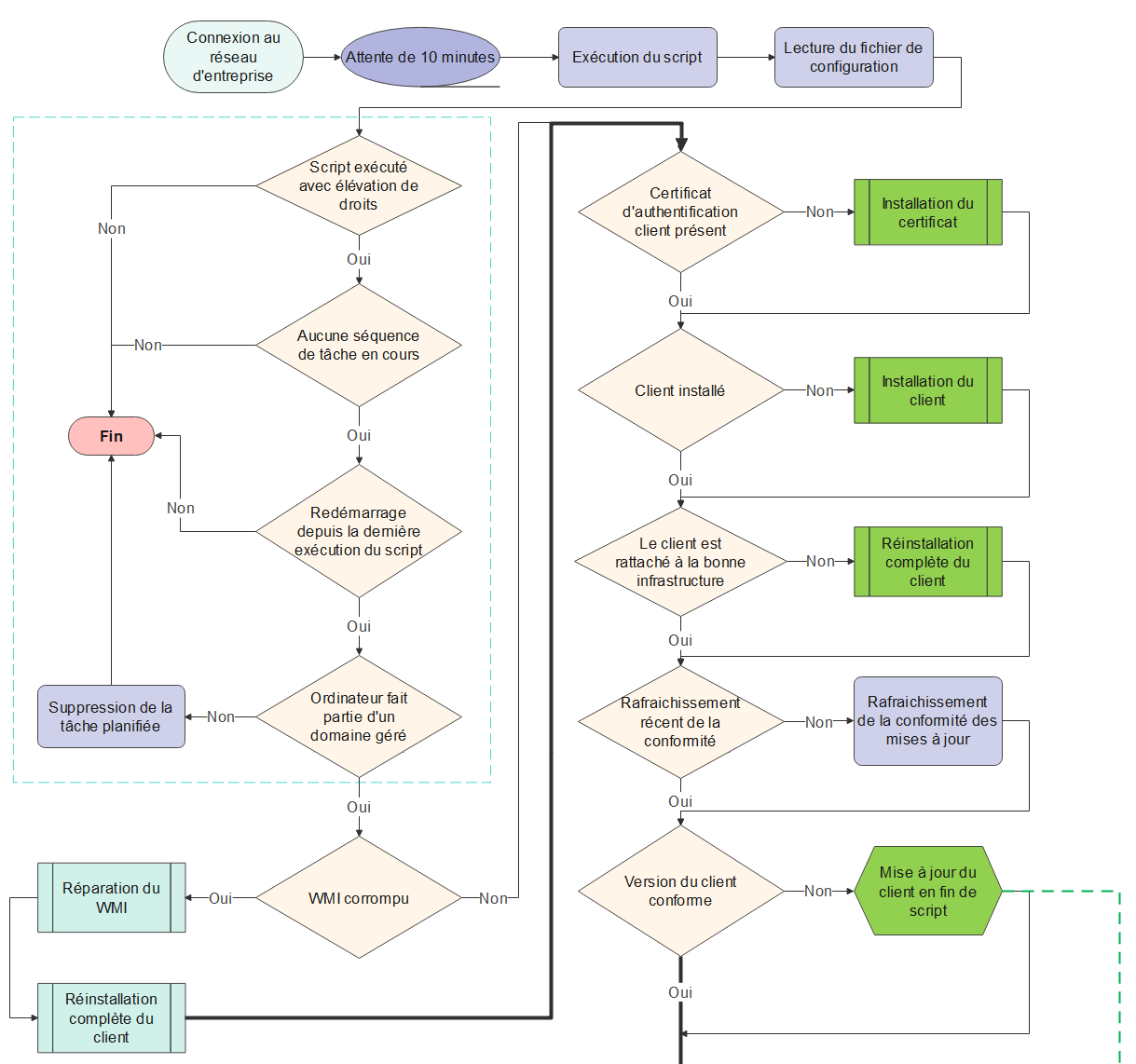
Déroulement du script exécuté par la GPO :

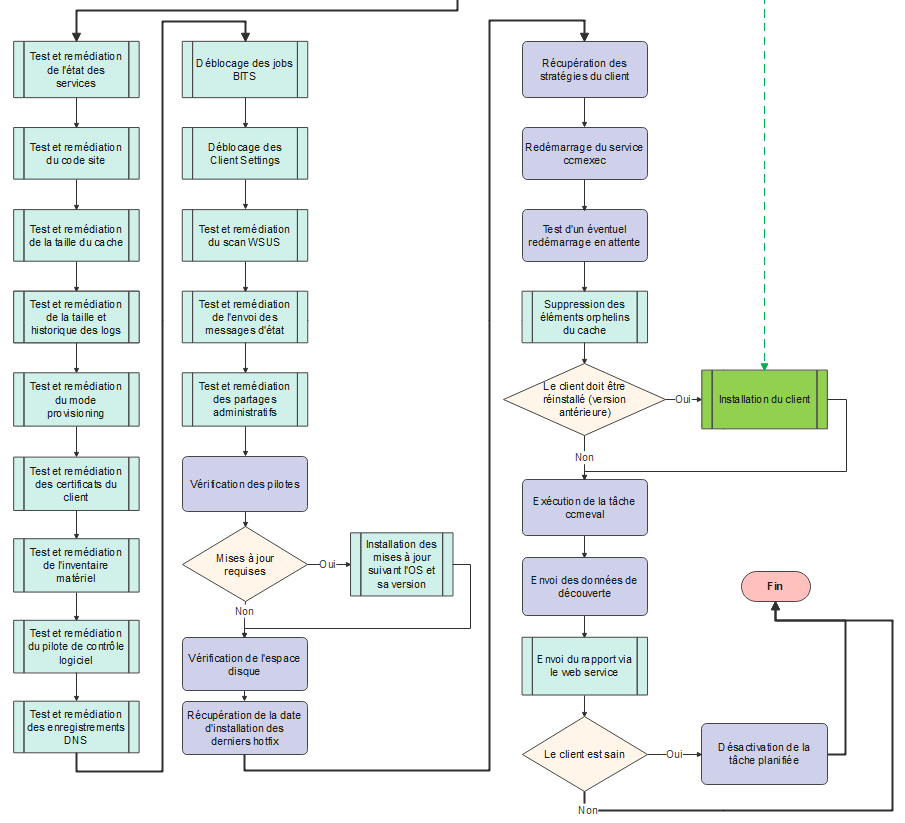


## Exécution du script

Le script s’exécute 10 minutes après la connexion au réseau d’entreprise (VPN inclus).

Ceci est possible en souscrivant à l'événement **ID 10000** du journal "**Microsoft-Windows-NetworkProfile/Operational**".





## Fichier de configuration

Le fichier de configuration défini les propriétés d’installation du client, les tests et remédiations à effectuer.

Ce fichier est au format xml et comprend les nœuds et balises suivantes.

Dans le cas où le client doit être installé ou réparé, cela se fera en utilisant les paramètres définis dans le nœud « ClientInstallProperty ».

### LocalFiles

Chemin d'accès à un répertoire local sur l'ordinateur exécutant le script, dans lequel le script stockera les fichiers temporaires (notamment les logs).

La valeur par défaut est C:\ProgramData\ConfigMgrClientHealth\Logs.

Valeur retenue

* Production : C:\ProgramData\ConfigMgrClientHealth\Logs
* Hors production : C:\ProgramData\ConfigMgrClientHealth\Logs

### Client

Paramètres à vérifier et à appliquer pour le client Configuration Manager.

#### Version

Version minimale du client Configuration Manager.

Un client qui exécute une version inférieure sera mis à niveau si l’attribut **AutoUpgrade** est égal à **True**.

Valeur retenue

* Production : 5.00.9096.1000 (Configuration Manager CB 2111)
* Hors production : 5.00.9096.1024 (Configuration Manager 2111 - KB16643863)

#### SiteCode

Code du site Configuration Manager.

Valeur retenue

* Production :
* Hors production :

#### Domain

Domaine des machines à traiter avec le script.

Valeur retenue

* Production :
* Hors production :

#### AutoUpgrade

Indique si le client doit être automatiquement mis à jour dans le cas où celui-ci se trouve dans une version inférieure à celle définie dans l’attribut **Version**.

Valeur retenue

* Production : False
* Hors production : False

#### Share

Emplacement des sources du client qui seront utilisées pour :

* Une nouvelle installation
* Une mise à niveau
* Une réinstallation dans le cadre de la remédiation

Cet emplacement doit être accessible par les comptes système des machines exécutant le script.

Si cet emplacement est un partage il est nécessaire de donner les droits SMB/NTFS en lecture aux groupes « Domain Computers » de chaque domaine.

Le script suivant peut être exécuté sur le serveur primaire de chaque infrastructure pour accomplir cette tâche :

[CmdletBInding()]

Param (

*# Folder containing the client sources*

    [String]$Path = 'D:\Microsoft Configuration Manager\Client',

*# Name of the share*

    [String]$Name = 'Client$',

*# Domain NETBIOS list*

    [String[]]$DomainList = ()

)

$GroupList = $DomainList | ForEach-Object {"$\_\Domain Computers"}

If ($null -eq (Get-SmbShare -Name $Name -ErrorAction Ignore)) {

    New-SmbShare -Path $Path -Name $Name -ReadAccess $GroupList -ContinuouslyAvailable $true

}

$ShareAccess = Get-SmbShareAccess -Name $Name | Select-Object -ExpandProperty AccountName

$acl = Get-Acl -Path $Path

foreach ($Group in $GroupList) {

    If ($ShareAccess -notcontains $Group) {

        Grant-SmbShareAccess -Name $Name -AccountName $Group -AccessRight Read -Force

    }

    If ($acl.Access.IdentityReference.Value -notcontains $Group) {

        $ace = New-Object System.Security.AccessControl.FileSystemAccessRule("$Group","ReadAndExecute", "ContainerInherit, ObjectInherit", "None", "Allow")

        $Acl.AddAccessRule($ace)

    }

}

Set-Acl -Path $Path -AclObject $acl

Valeur retenue

* Production : \\<Primary>\Client$
* Hors production : \\<Primary>\Client$

#### CacheSize

Défini la taille du cache requise par le client.

Attributs :

**Value** = Taille du cache en MB

**DeleteOrphanedData** = Suppression des éléments orphelins dans le cache

**Enable** = Indique si le paramètre doit être remédié si non conforme

Valeur retenue

* Production : Value="10240" DeleteOrphanedData="True" Enable="False"
* Hors production : Value="10240" DeleteOrphanedData="True" Enable="False"

#### Log

Défini la taille et l’historique des logs du client Configuration Manager à respecter.

Pour plus d’informations : [About log files](https://learn.microsoft.com/en-us/mem/configmgr/core/plan-design/hierarchy/about-log-files#client-and-management-point-logging-options)

Attributs :

**MaxLogSize** = Taille maximale des logs en MB

**MaxLogHistory** = Historique maximal des logs

**Enable** = Indique si le paramètre doit être remédié si non conforme

Valeur retenue

* Production : MaxLogSize="4096" MaxLogHistory="2" Enable="True"
* Hors production : MaxLogSize="4096" MaxLogHistory="2" Enable="True"

### ClientInstallProperty

Propriétés utilisées lors de l’installation du client Configuration Manager en utilisant ccmsetup.exe.

Il peut y avoir autant de balises « ClientInstallProperty » que voulu à partir du moment où la concaténation de leur valeur forme une ligne de commande conforme.

Pour plus d’informations : [Client installation parameters and properties](https://learn.microsoft.com/en-us/mem/configmgr/core/clients/deploy/about-client-installation-properties)

Valeur retenue

* Production :

SMSSITECODE=

SMSMPLIST=" "

/Source:C:\ProgramData\ConfigMgrClientHealth\CMClient

/UsePKICert

* Hors production :

SMSSITECODE=

SMSMPLIST=" "

/Source:C:\ProgramData\ConfigMgrClientHealth\CMClient

/UsePKICert

### Log

Paramètres liés aux fonctionnalités de journalisation et de rapport du script.

#### File

Déprécié

Valeur retenue

* Production : N/A
* Hors production : N/A

#### SQL

Serveur SQL à utiliser si les informations ne sont pas remontées via le WebService.

Non utilisé dans notre cas.

Valeur retenue

* Production : <SQLServer>
* Hors production : <SQLServer>

#### Time

Format de temps à utiliser dans les logs (ClientLocal ou UTC).

Valeur retenue

* Production : ClientLocal
* Hors production : ClientLocal

### Option

Paramètres du système à vérifier et à appliquer.

#### CcmSQLCELog

Vérifie la présence du log CcmSQLCE.log si le client n’est pas configuré en mode debug.

Pour plus d’informations : [Log file reference](https://learn.microsoft.com/en-us/mem/configmgr/core/plan-design/hierarchy/log-files#BKMK_ClientOpLogs)

Non utilisé.

Valeur retenue

* Production : False
* Hors production : False

#### BITSCheck

Permet de vérifier et de remédier les erreurs de tâches de téléchargements BITS.

Attributs :

**Fix** = Indique si le paramètre doit être remédié

**Enable** = Indique si le paramètre doit être vérifié et remonté dans les logs

Valeur retenue

* Production : Fix="True" Enable="True"
* Hors production : Fix="True" Enable="True"

#### ClientSettingsCheck

Vérifie la présence d’un blocage au niveau de la configuration client dans la base WMI et exécute une remédiation.

Le test recherche des stratégies bloquées de séquence de tâches (**PolicySource = CcmTaskSequence**) dans la classe **CCM\_ClientAgentConfig** du namespace « **root\ccm\Policy\DefaultMachine\RequestedConfig** ».

Attributs :

**Fix** = Indique si le paramètre doit être remédié

**Enable** = Indique si le paramètre doit être vérifié et remonté dans les logs

Valeur retenue

* Production : Fix="True" Enable="True"
* Hors production : Fix="True" Enable="True"

#### DNSCheck

Vérifie que les enregistrements DNS correspondent à l’IP de la machine et exécute une remédiation.

Attributs :

**Fix** = Indique si le paramètre doit être remédié

**Enable** = Indique si le paramètre doit être vérifié et remonté dans les logs

Valeur retenue

* Production : Fix="True" Enable="True"
* Hors production : Fix="True" Enable="True"

#### Drivers

Vérification de la présence de drivers non ou mal installés.

Aucune remédiation n’est possible.

Attributs :

**Enable** = Indique si le paramètre doit être vérifié et remonté dans les logs

Valeur retenue

* Production : Enable="True"
* Hors production : Enable="True"

#### Updates

Permet d’installer tous les correctifs manquants suivant la version du système d’exploitation.

Par exemple, un Windows 10 22H2 ira chercher ses mises à jour dans le dossier « **<Share>**\Windows 10 19045 ».

Attributs :

**Share** = Chemin où sont stockées les mises à jour

**Fix** = Indique si le paramètre doit être remédié

**Enable** = Indique si le paramètre doit être vérifié et remonté dans les logs

Valeur retenue

* Production : Share="" Fix="True" Enable="False"
* Hors production : Share="" Fix="True" Enable="False"

#### PendingReboot

Vérifie si un redémarrage est en attente pour les composants suivants :

* Component-Based Servicing
* Windows Update
* File rename pending
* Configuration Manager

Il est possible d’exécuter une application pour redémarrer l’ordinateur (Voir ci-dessous)

Attributs :

**StartRebootApplication** = Indique si l’application définie dans RebootApplication doit être utilisée

**Enable** = Indique si le paramètre doit être vérifié et remonté dans les logs

Valeur retenue

* Production : StartRebootApplication="False" Enable="True"
* Hors production : StartRebootApplication="False" Enable="True"

#### RebootApplication

Application à utiliser pour redémarrer l’ordinateur.

Non utilisé

Attributs :

**Application** = Chemin de l’application et ligne de commande

**Enable** = Indique si l’application peut être utilisée

Valeur retenue

* Production : Application="….\shutdowntool.exe /t:7200 /m:1440" Enable="False"
* Hors production : Application="….\shutdowntool.exe /t:7200 /m:1440" Enable="False"

#### MaxRebootDays

Défini le nombre de jour après redémarrage à partir duquel l’application **RebootApplication** doit être exécutée.

Non utilisé

Attributs :

**Days** = Nombre de jours d’activité

**Enable** = Aucune incidence

Valeur retenue

* Production : Days="7" Enable="False"
* Hors production : Days="7" Enable="False"

#### OSDiskFreeSpace

Vérifie le pourcentage d’espace disque libre sur le volume C en GB et remonte l’information dans les logs.

Aucune remédiation n’est possible.

Valeur retenue

* Production : 10
* Hors production : 10

#### HardwareInventory

Vérifie le nombre de jours depuis la dernière remontée d’inventaire matériel et exécute la remédiation.

Attributs :

**Days** = Nombre de jours depuis le dernier inventaire matériel après lequel relancer un inventaire

**Fix** = Exécute ou non la remédiation

**Enable** = Indique si le paramètre doit être vérifié et remonté dans les logs

Valeur retenue

* Production : Days="7" Fix="True" Enable="True"
* Hors production : Days="7" Fix="True" Enable="True"

#### SoftwareMetering

Vérifie l’état du pilote du contrôle logiciel.

Les messages suivants sont recherchés dans le log **mtrmgr.log** :

* **StartPrepDriver - OpenService Failed with Error**
* **Software Metering failed to start PrepDriver**

La remédiation est faite en exécutant :

**C:\Windows\system32\RUNDLL32.EXE SETUPAPI.DLL,InstallHinfSection DefaultInstall 128 C:\Windows\CCM\prepdrv.inf**

Pour plus d’informations : [Monitor app usage with software metering](https://learn.microsoft.com/en-us/mem/configmgr/apps/deploy-use/monitor-app-usage-with-software-metering)

Attributs :

**Fix** = Exécute ou non la remédiation

**Enable** = Indique si le paramètre doit être vérifié et remonté dans les logs

Valeur retenue

* Production : Fix="True" Enable="True"
* Hors production : Fix="True" Enable="True"

#### WMI

Vérifie la cohérence de la base WMI et exécute la remédiation.

Pour plus d’informations : [WMI: Rebuilding the WMI Repository](https://techcommunity.microsoft.com/t5/ask-the-performance-team/wmi-rebuilding-the-wmi-repository/ba-p/373846)

Attributs :

**Fix** = Exécute ou non la remédiation

**Enable** = Indique si le paramètre doit être vérifié et remonté dans les logs

Valeur retenue

* Production : Fix="True" Enable="True"
* Hors production : Fix="True" Enable="True"

#### RefreshComplianceState

Permet d’envoyer l’état de conformité des mises à jour de la machine au serveur de site si le dernier renvoi date d’un certain nombre de jours.

(New-Object -ComObject Microsoft.CCM.UpdatesStore).RefreshServerComplianceState()

Attributs :

**Days** = Nombre de jours après lequel renvoyer l’état de conformité

**Enable** = Indique si le paramètre doit être vérifié et remonté dans les logs

Valeur retenue

* Production : Days="7" Enable="True"
* Hors production : Days="7" Enable="True"

#### AllowCMG

Option ajoutée par rapport au script initial permettant l’utilisation de la CMG afin de récupérer les stratégies après l’installation du client.

Ce paramètre est normalement défini via un Client Settings « **Cloud Services -> Enable clients to use a cloud management gateway** » mais n’est pas disponible si le client n’est pas installé ou ne communique pas correctement.

Ce paramètre correspond à la valeur **DisAllowCMG** dans la clé **HKLM\SOFTWARE\Microsoft\CCM.**

La valeur est passée à 0 et la valeur **CMGFQDNs** est paramétrée avec le FQDN de la CMG.

Attributs :

**FQDN** = FQDN de la CMG (Ex : CMG.CONTOSO.COM)

**Enable** = Indique si le paramètre doit être vérifié et remonté dans les logs

Valeur retenue

* Production : FQDN="cmg.domain.com" Enable="True"
* Hors production : FQDN="" Enable="False"

#### ClientAuthCert

Option ajoutée par rapport au script initial permettant de vérifier la présence d’un certificat d’authentification client.

Les certificats sont recherchés dans le magasin personnel de l’ordinateur avec les critères suivants :

* Non expiré
* Issu d’un des templates renseignés
* Ayant un rôle « Authentification du client »

Si aucun certificat de ce type n’est trouvé, l’inscription est faite avec la commande suivante :

Get-Certificate -Template 'TemplateNameNoSpace' -CertStoreLocation 'Cert:\LocalMachine\My'

Attributs :

**Template** = Nom de templates séparés par une virgule

**Fix** = Exécute ou non la remédiation

**Enable** = Indique si le paramètre doit être vérifié et remonté dans les logs

Valeur retenue

* Production : Template=" " Fix="True" Enable="True"
* Hors production : Template=" " Fix="True" Enable="True"

### Service

Ces balises permettent de définir le type de démarrage et l’état de n’importe quel service installé.

Il est également possible de définir un nombre de jours d’activité après lequel le service doit être redémarré.

Attributs :

**Name** = Nom du service

**StartupType** = Type de démarrage (*Automatic, Automatic (Delayed Start), Disabled, Manual*)

**State** = Etat du service (*Running, Stopped*)

**Uptime** = Nombre de jours d’activité

Valeurs retenues

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DisplayName | Name | StartupType | State | UpTime |
| Service de transfert intelligent en arrière-plan | BITS | Automatic (Delayed Start) | Running |  |
| Infrastructure de gestion Windows | winmgmt | Automatic | Running |  |
| Windows Update | wuauserv | Automatic (Delayed Start) | Running |  |
| Serveur | lanmanserver | Automatic | Running |  |
| Appel de procédure distante (RPC) | RpcSs | Automatic | Running |  |
| Temps Windows | W32Time | Automatic | Running |  |
| SMS Agent Host | ccmexec | Automatic (Delayed Start) | Running |  |

### Remediation

#### AdminShare

Vérification des partages administratifs de la machine (Admin$ et C$).

Redémarre le service **lanmanserver** dans le cas où les partages ne sont pas présents.

Attributs :

**Fix** = Exécute ou non la remédiation

Valeur retenue

* Production : Fix="True"
* Hors production : Fix="True"

#### ClientProvisioningMode

Vérifie que le client ne se trouve pas en mode provisioning dans lequel ce dernier ne peut pas recevoir de stratégies.

La remédiation utilise la méthode **SetClientProvisioningMode** de la classe **SMS\_Client** du namespace **Root\ccm**

Pour plus d’informations : [Provisioning mode](https://learn.microsoft.com/en-us/mem/configmgr/osd/understand/provisioning-mode)

Attributs :

**Fix** = Exécute ou non la remédiation

Valeur retenue

* Production : Fix="True"
* Hors production : Fix="True"

#### ClientStateMessages

Vérifie que le client envoie correctement ses messages d’état.

La vérification porte sur le log **StateMessage.log** dans lequel le message suivant est recherché :

**Successfully forwarded State Messages to the MP**

La remédiation exécute :

(New-Object -ComObject Microsoft.CCM.UpdatesStore).RefreshServerComplianceState()

Attributs :

**Fix** = Exécute ou non la remédiation

Valeur retenue

* Production : Fix="True"
* Hors production : Fix="True"

#### ClientWUAHandler

Vérifie que  :

* C:\Windows\System32\GroupPolicy\Machine\registry.pol est mis à jour régulièrement.

Il est possible de préciser un nombre de jour après lequel le fichier sera considéré comme corrompu et la remédiation sera exécutée.

* Le journal d’événements **Microsoft-Windows-GroupPolicy/Operational** ne comporte pas d’erreur depuis la dernière exécution du script

Etapes de la remédiation :

* Suppession du fichier registry.pol
* Arrêt du service Windows Update
* Suppression du dossier C:\Windows\SoftwareDistribution
* Suppression du dossier C:\Windows\System32\catroot2
* Redémarrage du service Windows Update
* GPUpdate
* wuauclt.exe /resetauthorization /detectnow (resynchronization du client Windows Update)
* Redémarrage du service ccmexec
* Rafraichissement des stratégies du client

Attributs :

**Fix** = Exécute ou non la remédiation

**Days** = Nombre de jours depuis la dernière modification du fichier registry.pol à partir duquel considérer le fichier comme corrompu.

Valeur retenue

* Production : Fix="True" Days="30"
* Hors production : Fix="True" Days="30"

#### SMSCertificate

Vérifie que les certificats du client Configuration Manager sont bien enregistrés dans le magasin SMS.

La vérification porte sur le log **ClientIDManagerStartup.log** à la recherche des messages suivants :

* **Failed to find the certificate in the store** = Les certificats ne sont pas enregistrés
* **[RegTask] - Server rejected registration 3** = Le serveur rejette l’enregistrement du client

Etapes de la remédiation :

* Certificats non enregistrés => Arrêt du service **ccmexec** et suppression du fichier « **C:\ProgramData\Microsoft\Crypto\RSA\MachineKeys\19c5c\*** »
* Enregistrement rejeté => Aucune, la vérification doit être faite côté serveur.

Attributs :

**Fix** = Exécute ou non la remédiation

Valeur retenue

* Production : Fix="True"
* Hors production : Fix="True"

# Rapport

# Extension de console

Point d’attention : Ne pas abuser de l’utilisation de l’extension.

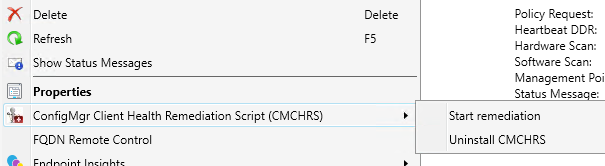
Avant de décider que le client a besoin d’être remédié, il est nécessaire de passer par des étapes de diagnostic pour savoir si le client communique ou s’il est simplement occupé.

L’extension est disponible depuis les serveurs rebond d’administration.

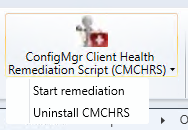
Cette dernière est affichée pour les éléments suivants :

* Devices
* Device Collections

On la retrouve dans le menu contextuel



Ainsi que dans le ruban



On dispose de 2 actions :

* **Start remediation**

Forcer l’installation de la solution et d’exécuter la tâche planifiée immédiatement.

* **Uninstall CMCHRS**

Désinstaller complètement la solution.

A chaque utilisation, si le domaine de l’utilisateur ne correspond pas au domaine de la machine cible, les identifiants de ce domaine seront demandés :

CAPTURE

Les identifiants sont sauvegardés pour ce domaine jusqu’à la fin du traitement de l’extension.

# Annexes

## Base de données

La base de données se nomme **ClientHealth** et a été créée sur les serveurs suivants :

* <ReportServer> (Production)
* <ReportServer> (Hors production)

La création se fait en utilisant le script suivant :

|  |
| --- |
| *-- SQL query to create and/or upgrade the database and tables for ConfigMgr Client Health*  *-- IF CREATING DATABASE FOR FIRST TIME:*  *-- Remember to grant 'domain\domain computers' DATAREADER and DATAWRITER rights on the ClientHealth database*  *-- after you execute this query and database is created.*  *-- START QUERY*  *-- Create database if not exist:*  GO  IF NOT EXISTS (SELECT [name] FROM sys.databases WHERE [name] = 'ClientHealth')  CREATE DATABASE ClientHealth  GO  USE ClientHealth  *-- Create Configuration table if not exist:*  GO  IF NOT EXISTS (SELECT [name] FROM sys.tables WHERE [name] = 'Configuration')  CREATE TABLE dbo.Configuration  (      Name varchar(50) NOT NULL UNIQUE,      Version varchar (10) NOT NULL  )  *-- Create Clients table if not exist:*  IF NOT EXISTS (SELECT [name] FROM sys.tables WHERE [name] = 'Clients')  CREATE TABLE dbo.Clients  (      Hostname varchar(100) NOT NULL PRIMARY KEY,      OperatingSystem varchar (100) NOT NULL,      Architecture varchar(10) NOT NULL,      Build varchar(50) NOT NULL,      Manufacturer varchar(50),      Model varchar(50),      InstallDate smalldatetime,      OSUpdates smalldatetime,      LastLoggedOnUser varchar(50),      ClientVersion varchar(20),      PSVersion float,      PSBuild int,      Sitecode varchar(3),      Domain varchar(50),      MaxLogSize int,      MaxLogHistory int,      CacheSize int,      ClientAuthCertificate varchar(50),      SMSCertificate varchar(50),      ProvisioningMode varchar(50),      DNS varchar(100),      Drivers varchar(100),      Updates varchar(100),      PendingReboot varchar(50),      LastBootTime smalldatetime,      OSDiskFreeSpace float,      Services varchar(50),      AdminShare varchar(50),      StateMessages varchar(50),      WUAHandler varchar(50),      WMI varchar(50),      RefreshComplianceState smalldatetime,      ClientInstalled smalldatetime,      Version varchar(10),      Timestamp datetime,      HWInventory smalldatetime,      SWMetering varchar(50),      BITS varchar(50),      PatchLevel int,      ClientInstalledReason varchar(200),      Compliant varchar(7)  )  else  *-- START Changes to database --*  *-- Add columns if needed*  IF NOT EXISTS (SELECT \* FROM sys.columns WHERE  object\_id = OBJECT\_ID(N'[dbo].[Clients]') AND name = 'HWInventory') ALTER TABLE dbo.Clients ADD HWInventory smalldatetime  IF NOT EXISTS (SELECT \* FROM sys.columns WHERE  object\_id = OBJECT\_ID(N'[dbo].[Clients]') AND name = 'SWMetering') ALTER TABLE dbo.Clients ADD SWMetering varchar(50)  IF NOT EXISTS (SELECT \* FROM sys.columns WHERE  object\_id = OBJECT\_ID(N'[dbo].[Clients]') AND name = 'BITS') ALTER TABLE dbo.Clients ADD BITS varchar(50)  IF NOT EXISTS (SELECT \* FROM sys.columns WHERE  object\_id = OBJECT\_ID(N'[dbo].[Clients]') AND name = 'PatchLevel') ALTER TABLE dbo.Clients ADD PatchLevel int  IF NOT EXISTS (SELECT \* FROM sys.columns WHERE  object\_id = OBJECT\_ID(N'[dbo].[Clients]') AND name = 'ClientInstalledReason') ALTER TABLE dbo.Clients ADD ClientInstalledReason varchar(200)  IF NOT EXISTS (SELECT \* FROM sys.columns WHERE  object\_id = OBJECT\_ID(N'[dbo].[Clients]') AND name = 'RefreshComplianceState') ALTER TABLE dbo.Clients ADD RefreshComplianceState smalldatetime  *-- Modify columns if needed*  IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Clients' AND COLUMN\_NAME = 'Hostname' AND DATA\_TYPE = 'varchar' AND CHARACTER\_MAXIMUM\_LENGTH = 100) ALTER TABLE dbo.Clients ALTER COLUMN Hostname VARCHAR(100) NOT NULL  IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Clients' AND COLUMN\_NAME = 'Build' AND DATA\_TYPE = 'varchar' AND CHARACTER\_MAXIMUM\_LENGTH = 100) ALTER TABLE dbo.Clients ALTER COLUMN Build VARCHAR(100) NOT NULL  IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Clients' AND COLUMN\_NAME = 'Manufacturer' AND DATA\_TYPE = 'varchar' AND CHARACTER\_MAXIMUM\_LENGTH = 100) ALTER TABLE dbo.Clients ALTER COLUMN Manufacturer VARCHAR(100) NOT NULL  IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Clients' AND COLUMN\_NAME = 'Model' AND DATA\_TYPE = 'varchar' AND CHARACTER\_MAXIMUM\_LENGTH = 100) ALTER TABLE dbo.Clients ALTER COLUMN Model VARCHAR(100) NOT NULL  IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Clients' AND COLUMN\_NAME = 'LastLoggedOnUser' AND DATA\_TYPE = 'varchar' AND CHARACTER\_MAXIMUM\_LENGTH = 100) ALTER TABLE dbo.Clients ALTER COLUMN LastLoggedOnUser VARCHAR(100) NOT NULL  IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Clients' AND COLUMN\_NAME = 'ClientVersion' AND DATA\_TYPE = 'varchar' AND CHARACTER\_MAXIMUM\_LENGTH = 100) ALTER TABLE dbo.Clients ALTER COLUMN ClientVersion VARCHAR(100) NOT NULL  IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Clients' AND COLUMN\_NAME = 'Drivers' AND DATA\_TYPE = 'varchar' AND CHARACTER\_MAXIMUM\_LENGTH = 100) ALTER TABLE dbo.Clients ALTER COLUMN Build VARCHAR(100) NOT NULL  IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Clients' AND COLUMN\_NAME = 'Domain' AND DATA\_TYPE = 'varchar' AND CHARACTER\_MAXIMUM\_LENGTH = 100) ALTER TABLE dbo.Clients ALTER COLUMN Domain VARCHAR(100) NOT NULL  IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Clients' AND COLUMN\_NAME = 'DNS' AND DATA\_TYPE = 'varchar' AND CHARACTER\_MAXIMUM\_LENGTH = 100) ALTER TABLE dbo.Clients ALTER COLUMN DNS VARCHAR(200) NOT NULL  IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Clients' AND COLUMN\_NAME = 'Updates' AND DATA\_TYPE = 'varchar' AND CHARACTER\_MAXIMUM\_LENGTH = 100) ALTER TABLE dbo.Clients ALTER COLUMN Updates VARCHAR(200) NOT NULL  IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Clients' AND COLUMN\_NAME = 'Services' AND DATA\_TYPE = 'varchar' AND CHARACTER\_MAXIMUM\_LENGTH = 100) ALTER TABLE dbo.Clients ALTER COLUMN Services VARCHAR(200) NOT NULL  *-- Set latest ConfigMgr Client Health database version:*  GO  begin tran  if exists (SELECT \* FROM dbo.Configuration WITH (updlock,serializable) WHERE Name='ClientHealth')  begin      IF EXISTS (SELECT \* FROM dbo.Configuration WITH (updlock,serializable) WHERE Name='ClientHealth' AND Version < '0.7.7')      UPDATE dbo.Configuration SET Version='0.7.7' WHERE Name = 'ClientHealth'  end  else  begin      INSERT INTO dbo.Configuration (Name, Version)      VALUES ('ClientHealth', '0.7.7')  end  commit tran  *-- End of query* |

La DB peut être mise à jour de 2 façons :

* Directement par les machines en utilisant leur compte système
* Via le web service en utilisant un compte de service

Dans notre cas, seul le web service est utilisé avec le compte de service :

* **<Account>** (Production)
* **<Account>** (Hors production)

Le compte utilisé pour interagir avec la DB doit avoir les rôles SQL suivants sur la base :

* db\_datareader
* db\_datawriter

## Web Service

Le web service a été installé sur chaque Management Point des infrastructures Configuration Manager afin de répartir les charges.

Le web service utilisé par chaque machine est défini lors de l’installation du script via GPO en se basant sur le domaine de celle-ci (voir Déploiement GPO).

### Prérequis

* Rôle IIS installé sur le serveur hôte du web service
* .NET Framework 4.8
* Base de données **ClientHealth** créée
* Un compte de service avec les rôles **db\_datareader** et **db\_datawriter** sur la base de données

### Installation

|  |  |
| --- | --- |
| Détails | Captures |
| Créer un dossier dans le dossier IIS (C:\inetpub) et le nommer « ConfigMgrClientHealth » |  |
| Extraire le contenu du fichier « ConfigMgr Client Health Webservice 2.0.1.zip » dans ce dossier |  |
| Editer le fichier web.config et changer « connectionString » avec le nom et l’instance du serveur SQL |  |
| Ouvrir la console IIS et créer un nouveau pool d’application |  |
| Le nommer « ConfigMgrClientHealth » et le paramétrer comme ci-contre |  |
| Aller dans les propriétés avancées afin de modifier le paramètre « Identity » avec le compte de service |  |
| Depuis le « Default Web Site », créer une application |  |
| La nommer « ConfigMgrClientHealth »  Utiliser le dossier et le pool créés précédemment |  |
| Redémarrer le pool |  |
| Vérifier le bon fonctionnement en allant sur l’url suivante :  server.domain/ConfigMgrClientHealth |  |

### Erreur 500

Lors de l’utilisation du script **ConfigMgrClientHealth.ps1**, si une erreur 500 apparait en fin de log, cela peut venir du fait que le compte de service n’est pas autorisé à accéder à la configuration du IIS.

Afin d’avoir plus d’informations sur l’erreur, il est possible d’activer les erreurs détaillées dans le fichier **web.config** en ajoutant cette partie dans le nœud **system.web** .

<configuration>

  <system.web>

**<customErrors mode="Off"/>**

  </system.web>

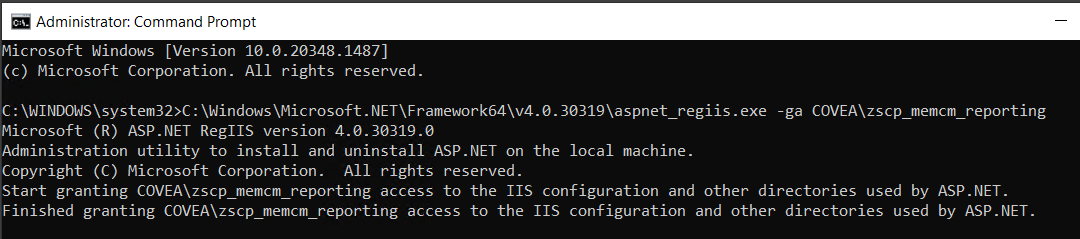
</configuration>

L’erreur suivante apparait ensuite en retour à la commande **Invoke-RestMethod** (fonction **Update-WebService**) :

**The current identity (<Domain>\<UserName>) does not have write access to 'C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\Temporary ASP.NET Files'**

Il suffit alors d’exécuter la commande suivante sur chaque serveur disposant du web service :

**C:\Windows\Microsoft.NET\Framework64\v4.0.30319\aspnet\_regiis.exe -ga *<Domain>\<UserName>***



## Extension de console

L’extension de console est installée en utilisant le script **Install-CMClientHealthConsoleExtension.ps1**.

Paramètres du script d’installation

|  |  |  |
| --- | --- | --- |
| Paramètre | Description | Valeur par défaut |
| Path | Chemin d’installation des binaires de l’extension (dll et script) | A préciser |
| ScheduledTaskName | Nom de la tâche planifiée créée sur les machines | A préciser |
| MaxThreads | Nombre de tâches parallèles maximales lorsque l’extension est utilisée sur un regroupement | 20 |
| SiteCode | Code site Configuration Manager | A préciser |

Les fichiers xml représentant l’extension sont modifiés par le script afin d’inclure les paramètres passés à celui-ci.

Ils sont ensuite copiés dans le dossier :

"$($ENV:SMS\_ADMIN\_UI\_PATH)\..\..\XmlStorage\Extensions\Actions".

$ENV:SMS\_ADMIN\_UI\_PATH correspond au dossier d’installation de la console Configuration Manager (Ex : C:\Program Files (x86)\Microsoft Endpoint Manager\AdminConsole\bin\i386)

Chaque nœud de console Configuration Manager correspond à un ID qui correspond au nom du dossier où mettre les fichiers xml :

**3fd01cd1-9e01-461e-92cd-94866b8d1f39** = Devices

**a92615d6-9df3-49ba-a8c9-6ecb0e8b956b** = Devices Collections

**ed9dee86-eadd-4ac8-82a1-7234a4646e62** = Devices Collections members

Les 2 actions disponibles utilisent le script **<ExtensionInstallPath>\Scripts\ConfigMgrClientHealthExtension.ps1** auquel sont passés plusieurs paramètres :

|  |  |
| --- | --- |
| Paramètre | Description |
| Type | Device ou Collection suivant le nœud à partir duquel l’action est exécutée |
| ResourceId | ID de la ressource (Device ou Collection) passée par la variable **##SUB :ResourceID##** |
| MaxThread | Nombre de tâches parallèles maximales lorsque l’extension est utilisée sur un regroupement (20) |
| TaskName | Nom de la tâche planifiée créée sur les machines |
| SiteCode | Code site Configuration Manager |
| Install (non utilisé) | Switch indiquant l’action à réaliser (Installation du script de remédiation sur la machine) |
| Start | Switch indiquant l’action à réaliser (Installation et exécution du script de remédiation sur la machine) |
| Uninstall | Switch indiquant l’action à réaliser (Désinstallation du script de remédiation sur la machine) |

Le FQDN des machines est récupéré en utilisant l’admin service Configuration Manager d’où le passage du code site en paramètre pour récupérer le FQDN du « SMS Provider » du site.

Le FQDN du serveur ne peut pas être récupéré en utilisant la variable **##SUB :\_\_Server##** car elle ne contient que son nom NetBIOS.

Documentations traitant de la création d’extensions de console :

[Create a Configuration Manager Action](https://learn.microsoft.com/en-us/mem/configmgr/develop/core/servers/console/how-to-create-a-configuration-manager-action)

[Create Your Own Right Click Tools](https://www.ephingadmin.com/create-your-own-right-click-tools/)

## Validation de l’installation du script

|  |  |
| --- | --- |
| Détails | Captures |
| Vérifier que la machine n’est pas présente dans le regroupement « LDC CMClients SystemHealth RemediationExclusion » | |
| Après un premier redémarrage, les sources sont présentes dans C:\ProgramData\ConfigMgrClientHealth |  |
| La tâche est présente après le redémarrage suivant |  |
| Connexion au réseau de l’entreprise |  |
| Tâche inscrite mais pas de déclenchement car pas de tâche au moment de la connexion |  |
| Déclenchement de la tâche exactement 10 minutes après la connexion au réseau d’entreprise |  |
| Filtre XPath à utiliser dans l’observateur d’événements pour voir les connexions au réseau d’entreprise et les événements liées à la tâche dans une même vue   |  | | --- | | <QueryList>      <Query Id="0" Path="Microsoft-Windows-TaskScheduler/Operational">              <Select Path="Microsoft-Windows-TaskScheduler/Operational">              \*[EventData[Data[@Name="TaskName"]="\ConfigMgr Client Health Remediation Script"]]          </Select>      </Query>      <Query Id="1" Path="Microsoft-Windows-NetworkProfile/Operational">          <Select Path="Microsoft-Windows-NetworkProfile/Operational">              \*[System[                  Provider[@Name='Microsoft-Windows-NetworkProfile'] and (EventID=10000)              ]              and EventData[                  Data[@Name='Name']!='Identification...'                  and Data[@Name='Type']=1]              ]          </Select>      </Query>  </QueryList> | | |
| Log créé dans le dossier C:\ProgramData\ConfigMgrClientHealth\Logs |  |
| Fin du log avec le message « Updating SQL database with results using webservice » ce qui signifie que la communication est bonne entre le client et le web service ainsi qu’entre le web service et la base de données.  Les avertissements qui suivent montrent que le client n’est pas encore considéré comme conforme.  Si ces tests sont conformes à la prochaine exécution du script, la tâche planifiée sera désactivée |  |
| A l’exécution suivante, les avertissements ont disparu et la tâche planifiée est désactivée |  |