



# SpnClient

## MANUEL UTILISATEUR

---

20/03/2015

## TABLE DES MATIERES

<b>1. Introduction .....</b>	<b>6</b>
1.1. But du document .....	6
1.2. Version .....	6
1.3. Conventions typographiques .....	6
<b>2. Fichier de configuration .....</b>	<b>7</b>
2.1. Syntaxe .....	7
2.2. Nom du fichier .....	7
2.3. Edition du fichier .....	7
2.4. Localisation et prise en compte .....	7
2.5. Sections dynamiques .....	8
<b>3. Options de configuration .....</b>	<b>9</b>
3.1. Loader .....	9
3.1.1. LauncherMode .....	9
3.1.2. WdpStats .....	9
3.2. Application .....	9
3.2.1. WdpView .....	9
3.2.2. RcPopUp .....	10
3.2.3. Language .....	10
3.3. Connection .....	10
3.3.1. ServerIp .....	10
3.3.2. ServerPort .....	10
3.3.3. LockOnServer .....	11
3.3.4. ConnType .....	12
3.3.5. GprsConnMode .....	12
3.3.6. GprsConnName .....	12
3.3.7. GprsWindowColor .....	13
3.3.8. GprsShowWindow .....	13
3.3.9. ConnectionMode .....	13
3.3.10. ConnMgr_DestinationNet .....	13
3.4. Buildvpn .....	14
3.4.1. VPNBuildConfig .....	14
3.4.2. VPNConnName .....	14
3.4.3. VPNserverIp .....	14
3.4.4. VPNPublicIpAddress .....	14
3.4.5. VPNUserName .....	14
3.4.6. VPNPassword .....	15
3.5. Initinterfaces .....	16
3.5.1. GPRS .....	16
3.5.2. Bluetooth .....	16
3.5.3. Wifi .....	16
3.6. Terminal .....	17
3.6.1. UserId .....	17
3.6.2. SetHostName .....	18
3.6.3. ExtendedInfo .....	19
3.6.4. ExtendedInfoOnUpdate .....	19
3.6.5. PhoneLineInfo .....	20
3.6.6. OemInfo .....	21
3.7. RegKeys .....	21

3.7.1. RegKeyCount .....	22
3.7.2. RegKeyXX .....	22
3.8. XmlFiles .....	23
3.8.1. XmlKeyCount.....	23
3.8.2. XmlXX.....	23
3.9. Stats .....	25
3.9.1. CipherStats .....	25
3.9.2. CipherKey .....	25
3.9.3. SendStats .....	25
3.9.4. SendVxtLog .....	25
3.10. Time .....	26
3.10.1. SyncTimeWdp .....	26
3.10.2. SyncTimeNtp .....	26
3.10.3. TimeZone .....	26
3.10.4. ServerNtp .....	26
3.10.5. PeriodNtp.....	26
3.11. AutoUpdate .....	27
3.11.1. DoStartUpCommand .....	27
3.11.2. StartUpCommand.....	27
3.11.3. StartUpCommandTimeout.....	27
3.11.4. DoPeriodic .....	29
3.11.5. Period .....	29
3.11.6. PeriodicCommand .....	29
3.11.7. PeriodicWakeUpDevice.....	29
3.11.8. DoSchedule .....	29
3.11.9. ScheduleCommand .....	30
3.11.10. OnConnect .....	30
3.11.11. OnConnectInterval.....	30
3.11.12. OnConnectCommand.....	31
3.11.13. doWeekActivity .....	31
3.11.14. weekActivity .....	31
3.12. Security .....	33
3.12.1. doMonitorDeadServer .....	33
3.12.2. deadServerTimeout.....	34
3.12.3. deadServerScript.....	34
3.12.4. repeatDeadServerScript.....	34
3.13. Schedule .....	35
3.13.1. ConnCount .....	35
3.13.2. ConnStartXX.....	35
3.13.3. ConnEndXX.....	36
3.13.4. ConnPeriodXX.....	36
3.13.5. ConnWakeUpDeviceXX .....	36
3.14. ScheduleScript .....	37
3.14.1. Count .....	37
3.14.2. TimeXX.....	37
3.14.3. ScriptXX.....	37
3.15. Deployment .....	38
3.15.1. SkipOnError.....	38
3.15.2. SendUpdateLog.....	38
3.15.3. SilentDeployment .....	38
3.15.4. AskToInstall .....	38
3.15.5. AskToInstallNo .....	38
3.15.6. WarmBootExe .....	39
3.15.7. ColdBootExe.....	39

<b>4. Gestion du démarrage</b>	<b>40</b>
4.1. Exécution automatique de scripts	40
<b>5. Syntaxe des scripts wdp</b>	<b>41</b>
5.1. MAIN	42
5.1.1. delayBeforeRunning	42
5.1.2. sequentialActions	42
5.1.3. autodelete	43
5.2. ACTION-ZIP	43
5.2.1. FileCount	44
5.2.2. Stoponerror	44
5.2.3. FileXX	44
5.3. ACTION-CPY	44
5.3.1. FileCount	44
5.3.2. stoponerror	45
5.3.3. FileXX	45
5.4. ACTION-ATTRIB	45
5.4.1. FileCount	45
5.4.2. stoponerror	45
5.4.3. FileXX	46
5.5. ACTION-REG	46
5.5.1. FileCount	47
5.5.2. stoponerror	47
5.5.3. FileXX	47
5.6. ACTION-WAIT-EXE	47
5.6.1. FileCount	47
5.6.2. stoponerror	47
5.6.3. FileXX	47
5.7. ACTION-EXE	48
5.7.1. FileCount	48
5.7.2. stoponerror	48
5.7.3. FileXX	48
5.8. ACTION-REGEXP	48
5.8.1. FileCount	48
5.8.2. stoponerror	48
5.8.3. FileXX	49
5.9. ACTION-DEL	49
5.9.1. FileCount	49
5.9.2. stoponerror	49
5.9.3. FileXX	49
5.10. ACTION-REMOVE	49
5.10.1. FileCount	50
5.10.2. stoponerror	50
5.10.3. FileXX	50
5.11. ACTION-KILL	50
5.11.1. FileCount	50
5.11.2. stoponerror	50
5.11.3. FileXX	50
5.12. ACTION-DELAY	50
5.12.1. delay	50

5.13. ACTION-BOOT .....	51
5.13.1. type .....	51
5.13.2. immediate .....	51
5.13.3. delay .....	51
<b>6. Recommandations pour le paramétrage.....</b>	<b>52</b>
6.1. Déploiements automatiques .....	52
6.2. Optimisation des commandes périodiques .....	52
6.3. Répertoire d'installation .....	52

### Historique du document:

Révision	Auteur(s)	Modifications
1	Sandrine MARTINEZ Frédéric SACCOCCIO	Révision initiale
2	Frédéric SACCOCCIO	Ajout d'un paragraphe pour les recommandations de configuration.
3	Frédéric SACCOCCIO	Correction de l'option language de la section [application], le paramétrage de l'espagnol se fait avec la valeur sp et non es
4	Frédéric SACCOCCIO	Ajout des paragraphes 5 et 6 sur les scripts .wdp
5	Frédéric SACCOCCIO	<ul style="list-style-type: none"><li>▪ Renumérotation des paragraphes (les recommandations pour le paramétrage sont en maintenant en dernier).</li><li>▪ Corrections sur le contenu des scripts wdp</li></ul>
6	Frédéric SACCOCCIO	<ul style="list-style-type: none"><li>▪ Changement de titre</li><li>▪ Ajout des paragraphes sur la gestion des plages d'activité</li><li>▪ Ajout des paragraphes sur la détection d'un événement "Dead server"</li><li>▪ Ajout de précision sur le paramètre TimeZone de la section Time</li></ul>
7	Frédéric SACCOCCIO	<ul style="list-style-type: none"><li>▪ Ajout du paramètre LockOnServer dans la section Connection.</li></ul>
8	Frédéric SACCOCCIO	<ul style="list-style-type: none"><li>▪ Amélioration de la syntaxe du paramètre SetHostName de la section [Terminal]</li></ul>

## 1. Introduction

SpnClient est un logiciel présent dans la suite de logiciels du client Symphony. Il s'agit du client Symphony lui-même embarqué sur les terminaux mobiles. A ce titre c'est, lui qui communique avec la Console Symphony.

### 1.1. But du document

Afin de pouvoir être adaptable et de répondre à la plupart des exigences, SpnClient est entièrement paramétrable. Le présent document expose les différentes options disponibles et explique comment les configurer. Ce document s'adresse donc aux personnes en charge de paramétrer et configurer SpnClient (chefs de projets, responsables d'exploitation).

### 1.2. Version

Cette documentation concerne le binaire SpnClient.exe dans ses versions **5.2.5.0** ou supérieures. Ce module logiciel fait partie du paquet d'installation du client Symphony pour Windows CE/Mobile version **2.2.0**.

### 1.3. Conventions typographiques

Il est possible de naviguer dans ce document à l'aide d'hyperliens. Ces derniers sont indiqués par la typographie suivante : ***hyperlien***.

Les mots ou éléments importants seront en gras et en italique comme ce ***mot*** pour être mis en évidence.

Les nouveautés qui apparaissent à partir d'une certaine version sont indiquées comme suit:

new

**Version 3.2.5.0**

## 2. Fichier de configuration

### 2.1. Syntaxe

Le fichier de configuration de SpnClient est un fichier au format Microsoft INI. Il s'agit donc d'un fichier de type texte avec un encodage ASCII.

Un fichier INI est composé d'un ensemble de **sections**.

Une **section** débute par le nom de la section entre crochets sur une ligne suivie des **éléments** de la section. Chaque élément est une ligne de la forme

```
launcherMode=yes
```

La partie à gauche du signe égal est le **nom** de l'élément et la partie à droite du signe égal est la **valeur**.

### Voici un exemple de fichier INI avec 2 sections (loader et application)

```
[loader]
launcherMode=yes
wdpStats=no

[application]
wdpView=yes
rcPopUp=A remote control session is being started by the Wdp Console Administrator.
language=fr
```

### 2.2. Nom du fichier

Le fichier de configuration de SpnClient est nommé SpnClient.cfg

### 2.3. Edition du fichier

L'édition du fichier de configuration de SpnClient doit être faite avec l'outil IniEditor.NET qui est fourni dans le paquet d'installation du client Symphony.

### 2.4. Localisation et prise en compte

Le fichier de configuration de SpnClient doit se trouver dans le même répertoire que l'exécutable SpnClient.exe. Le paquet d'installation du client Symphony contient tous les outils nécessaires pour déployer l'ensemble des logiciels du client Symphony de façon à assurer que SpnClient.exe et SpnClient.cfg soient bien dans le même répertoire.

Il est pris en compte au démarrage de SpnClient. Si un élément de ce fichier est changé alors que SpnClient est lancé alors l'application devra être redémarrée pour la prise en compte du changement.



## 2.5. Sections dynamiques

Certaines sections du fichier de configuration sont qualifiées de **dynamiques**. Une section **dynamique** est une section où un lot d'éléments se répète. Un élément particulier (le compteur) indique le nombre de répétitions. Cet élément a un nom qui a pour suffixe "Count" et il doit être le premier élément de la section. Voici un exemple :

```
FileCount=1
```

Chacun des éléments d'un lot a pour suffixe le numéro de ce lot sur 2 digits décimaux.

Voici ci-dessous un exemple complet de section dynamique :

```
[tools2]
FileCount=2
startExe00=\Windows\Clock.exe
name00=Clock
startExe01=\Windows\calendar.exe
name01=Calendar
```

Dans l'exemple ci-dessus, la section comporte 2 répétitions et le lot d'éléments est composé des noms suivants :

- startExe**XX**
- name**XX**

où **XX** représente l'index du lot.

Les index sont numérotés sur 2 digits décimaux à partir de 00 (zéro).

L'édition des sections dynamiques est particulièrement facilitée par l'outil IniEditor.NET.

### Note

*Une section dynamique sera indiquée par la mention suivante dans le reste de ce document :*



**SECTION DYNAMIQUE**

Une section dynamique sera spécifiée dans un tableau comme ci-dessous :

Élément	Définition
Compteur	NomDuCompteur
Lot	<ul style="list-style-type: none"> <li>▪ ElementLot_01_XX</li> <li>▪ ElementLot_02_XX</li> <li>▪ ElementLot_03_XX</li> <li>▪ ...</li> <li>▪ ElementLot_nn_XX</li> </ul>

La partie **XX** dans la description du lot commence à 00 et sera ensuite incrémentée pour chaque occurrence suivante du lot au sein de la section : 00, 01, 02, 03, etc...

## 3. Options de configuration

Les paragraphes suivants décrivent toutes les sections du fichier de configuration ainsi que leur contenu. Les paragraphes de niveau 2 (numérotés 3.x) représente les sections tandis que leur sous-paragraphes (numéroté 3.x.y) représente les éléments de la section.

### 3.1. Loader

Cette section concerne le lancement de 2 autres applications qui font partie du client Symphony:

- SpnUserBox : le Kiosque du client Symphony
- WdpStats : le module de statistiques du client Symphony

#### 3.1.1. LauncherMode

Cette option permet d'activer ou de désactiver le lancement automatique de SpnUserBox au démarrage de l'appareil.

Valeurs possibles	Signification
yes	Lance automatiquement SpnUserBox au démarrage de l'appareil
no	Désactive le lancement automatique de SpnUserBox lors du démarrage de l'appareil

#### 3.1.2. WdpStats

Cette option permet d'activer ou de désactiver le lancement automatique de WDPstats au démarrage de l'appareil.

Valeurs possibles	Signification
yes	WDPstats.exe est <b>lancé</b>
no	WDPstats.exe <b>N'EST PAS lancé</b>

## 3.2. Application

### 3.2.1. WdpView

Cette option permet de contrôler l'affichage des messages durant l'installation des fichiers ".wdp" au démarrage du client Symphony.

Si cette option n'est pas activée l'installation se fait en arrière-plan.

Valeurs possibles	Signification
yes	Active l'affichage des messages durant l'installation des fichiers .wdp au démarrage de l'application
no	Aucun message n'est affiché au démarrage pour l'installation des scripts .wdp.

#### 3.2.2. RcPopUp

Personnalise le message de la notification (pop up) qui s'affiche à l'écran du PDA lorsqu'une session de contrôle à distance du PDA est lancée par la console d'administration.

Valeurs possibles	Signification
Chaîne de caractères	Définit le message qui va apparaître dans la pop-up informant de la prise en main à distance du pda.

**Exemple :**

```
rcPopUp=A remote control session is being started by the Wdp Console Administrator
```

#### 3.2.3. Language

Permet de choisir la langue utilisée pour afficher les messages à l'utilisateur.

Valeurs possibles	Signification
en	Définit la langue en anglais
fr	Définit la langue en français
sp	Définit la langue en espagnol

### 3.3. Connection

#### 3.3.1. ServerIp

Option permettant de définir le serveur auquel doit être relié le PDA en renseignant son adresse IP en notation décimale pointée.

Valeurs possibles	Signification
Adresse IP	Adresse IP du serveur auquel est relié le PDA en notation décimale pointée

**Exemple :**

```
serverIp=172.21.2.33
```

#### 3.3.2. ServerPort

Renseigne le numéro de port TCP/IP du serveur.

Valeurs possibles	Signification
Entier entre 1 et 65535	Numéro de port TCP/IP du serveur

### 3.3.3. LockOnServer

**new** Version 5.2.5.0

Ce paramètre permet de contrôler le comportement du client Symphony quand il reçoit une des commandes suivantes:

- LIST
- UPDATE
- RESET

En effet, chacune de ces commandes va entraîner une (ou plusieurs) requête(s) du client vers **une** console Symphony. Et par défaut, si les paramètres (adresse IP et port TCP/IP) d'une console Symphony sont inclus dans la commande alors le client va s'adresser à cette console. Il y a donc un risque que le terminal réponde à une console qui n'est pas forcément sa console dédiée (c'est notamment le cas dans l'utilisation de WireLess Deployer LocConsole).

Le paramètre LockOnServer permet de gérer de façon sécurisée le comportement du client face aux commandes qu'il reçoit : si LockOnServer est positionné à yes alors le client Symphony va systématiquement vérifier si les paramètres de console présents dans la commande sont ceux du fichier de configuration et n'exécutera la commande que s'il y a concordance.

Il est rappelé que si la commande ne comporte pas les paramètres de la console à cibler alors ce sont les paramètres du fichier de configuration (SpnClient.cfg) qui sont utilisés.

Valeurs possibles	Signification
yes	Le client Symphony <b>n'exécute pas</b> les commandes LIST, UPDATE ou RESET si ces dernières contiennent des paramètres (adresse IP et port TCP/IP) de console <b>différents des paramètres stockés</b> dans le fichier de configuration.
no	Le client <b>exécute</b> les commandes LIST, UPDATE ou RESET <b>même si ces dernières contiennent des paramètres (adresse IP et port TCP/IP) de console différents</b> des paramètres stockés dans le fichier de configuration.

#### 3.3.4. ConnType

Type de connexion utilisée pour se connecter au serveur.

Valeurs possibles	Signification
ethernet	Utilise la connexion de type Ethernet ou Wifi
gprs	Utilise la connexion de type GPRS

#### 3.3.5. GprsConnMode

Mode de numérotation et fin des connexions GPRS.

La connexion utilisée est celle dont le nom est stocké dans le paramètre [GprsConnName](#) ci-après.

Valeurs possibles	Signification
DH	Numérote puis raccroche à la fin de la connexion seulement si le client Symphony a réalisé la numérotation. Si la connexion GPRS se trouve déjà établie, le client Symphony ne raccroche pas.
DO	Numérote mais ne raccroche pas à la fin de la connexion.
CDH	Raccroche toute connexion existante. Puis numérote et raccroche à la fin de la connexion seulement si le client Symphony a réalisé la numérotation. Si la connexion GPRS se trouve déjà établie, le client Symphony ne raccroche pas.
CDO	Raccroche toute connexion existante. Puis numérote et ne raccroche pas.
NDNH	Ne numérote pas et ne raccroche pas.

#### 3.3.6. GprsConnName

Fourni le nom de la connexion utilisée pour accéder au réseau GPRS.

Valeurs possibles	Signification
Chaîne de caractères	Donne le nom de la connexion réseau GPRS utilisée

#### 3.3.7. GprsWindowColor

Personnalise la couleur de fond de la fenêtre de connexion GPRS

Valeurs possibles	Signification
Code hexadécimal d'une couleur au format RGB	Définit la couleur de fond de la fenêtre de connexion GPRS

La syntaxe est la suivante : 0xRRGGbb où

- **RR** : la composante rouge sur un octet en notation sur 2 digits hexadécimaux,
- **GG** : la composante verte sur un octet en notation sur 2 digits hexadécimaux
- **bb** : la composante bleue sur un octet en notation sur 2 digits hexadécimaux

**Exemple :**

```
gprsWindowColor=0x8080C8
```

#### 3.3.8. GprsShowWindow

Gère l'affichage de la fenêtre de connexion GPRS.

Valeurs possibles	Signification
yes	Affiche la fenêtre de connexion GPRS
no	Masque la fenêtre de connexion GPRS

#### 3.3.9. ConnectionMode

Choix du mode de connexion GPRS.

Valeurs possibles	Signification
RasDial	Méthode traditionnelle permettant seulement des connexions exclusives.
ConnMgr	Le Connection Manager permet de partager des connexions GPRS entre logiciel.

#### 3.3.10. ConnMgr\_DestinationNet

Définit le réseau de destination pour la Connection Manager.

Valeurs possibles	Signification
Work	Utilise la connexion du profil Work
MyISP	Utilise la connexion FAI (fournisseur d'accès)

#### 3.4. Buildvpn

Configure les options du client pour le VPN.

##### **Note**

*Cette section ne concerne que les systèmes d'exploitation de type Windows Mobile ou Windows Embedded Handheld.*

##### 3.4.1. VPNBuildConfig

Crée la configuration VPN à partir des options VPNConnName, VPNPublicIpAddress, VPNUserName et VPNPassword.

La connexion VPN est configurée grâce à l'option **ConnMgr\_DestinationNet** et est de type PPTP (Point-to-point tunneling protocol).

Valeurs possibles	Signification
yes	Active le réseau VPN
no	Désactive le réseau VPN

##### 3.4.2. VPNConnName

Renseigne le nom de la connexion VPN.

Valeurs possibles	Signification
Chaîne de caractères	Nom de la connexion VPN

##### 3.4.3. VPNserverIp

Renseigne l'adresse IP privée assignée au serveur VPN en notation décimale pointée.

Valeurs possibles	Signification
Adresse IP	Donne l'adresse IPv4 privée du serveur VPN en notation décimale pointée

##### 3.4.4. VPNPublicIpAddress

Renseigne l'adresse IP publique assignée au serveur VPN en notation décimale pointée.

Valeurs possibles	Signification
Adresse IP	Donne l'adresse IPv4 publique du serveur VPN en notation décimale pointée

##### 3.4.5. VPNUserName

Renseigne le nom d'utilisateur pour accéder à la connexion VPN.

Valeurs possibles	Signification
Chaîne de caractères	Nom d'utilisateur pour se connecter à la connexion VPN

### 3.4.6. VPNPassword

Renseigne le mot de passe pour accéder à la connexion VPN.

<b>Valeurs possibles</b>	<b>Signification</b>
Chaîne de caractères	Mot de passe pour se connecter à la connexion VPN



### 3.5. Initinterfaces

Configure les options du client pour l'initialisation des interfaces wireless (Wifi, Bluetooth, WWAN).

 **Attention**

*Les options de cette section ne sont pas implémentées.*

#### 3.5.1. GPRS

 **Attention**

*Cette option n'est pas implémentée.*

Valeurs possibles	Signification
skip	
yes	
no	

#### 3.5.2. Bluetooth

 **Attention**

*Cette option n'est pas implémentée.*

Valeurs possibles	Signification
skip	
yes	
no	

#### 3.5.3. Wifi

 **Attention**

*Cette option n'est pas implémentée.*

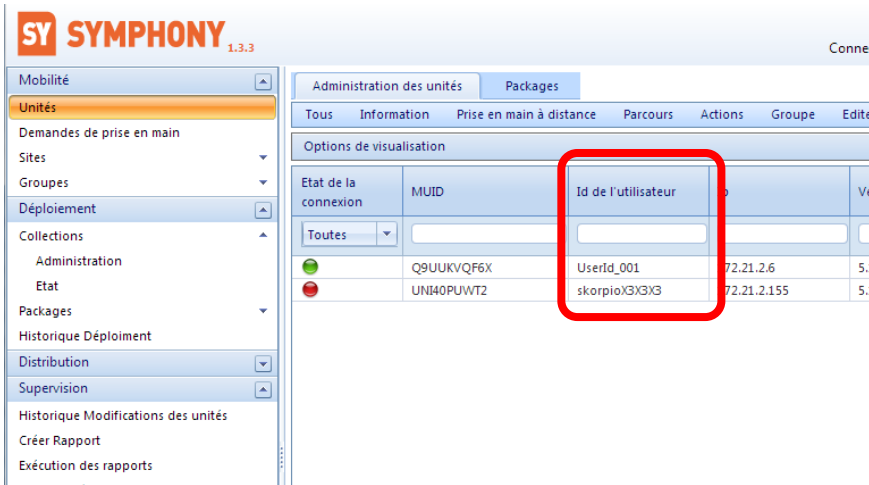
Valeurs possibles	Signification
skip	
yes	
no	

### 3.6. Terminal

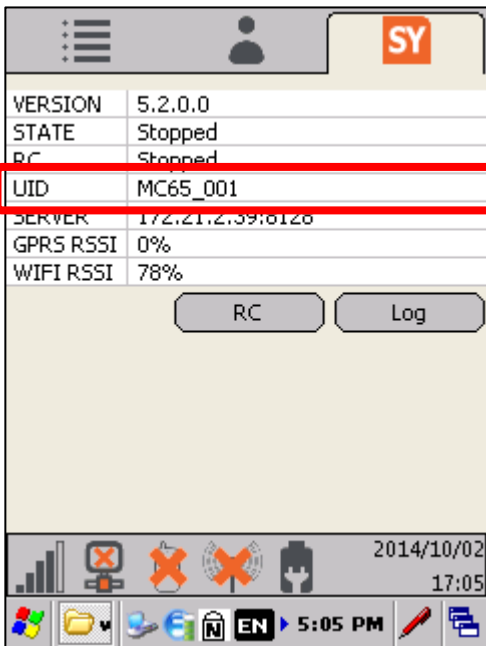
Configure les options liées au terminal.

#### 3.6.1. UserId

Renseigne un nom pour le terminal afin de l'identifier dans Symphony. Ce nom est visible dans la Console Symphony (Section Unités → Colonne "Id de l'utilisateur") comme illustré ci-dessous :



Cet identifiant est aussi visible dans SpnUserBox dans l'onglet Symphony comme illustré ci-après (champ UID) :



Valeurs possibles	Signification
Chaine de caractères	Nom donné au terminal

#### 3.6.2. SetHostName

Ce paramètre permet de changer le nom de l'appareil (Hostname) du PDA. Ce nom est une caractéristique de Windows CE/Mobile et n'est pas propre à Symphony comme l'est le UserId précédent.

La valeur de ce champ peut contenir des variables qui sont remplacées par une valeur particulière. Voici la liste des variables concernées ainsi que leur valeur :

Variables	Signification
(MUID)	MUID du terminal
(MAC)	Adresse MAC du terminal (les 6 octets de l'adresse MAC sont séparés par des underscores)
(USERID)	UserId de Symphony

Il est possible de conserver la valeur actuelle stockée dans le terminal avec la syntaxe suivante :  
setHostName=(none)

#### Voici quelques exemples :

```
setHostname=EDA_(MUID)
```

En supposant que le MUID du terminal soit I0DEWGKBK alors le hostname du terminal aura pour valeur finale :  
EDA\_I0DEWGKBK

```
setHostname=(MAC)_PDA
```

En supposant que l'adresse MAC du terminal soit 00:16:9D:7C:C7:50 alors le hostname du terminal aura pour valeur finale :

```
00_16_9D_7C_C7_50_PDA
```



#### Attention !

*Il est fortement déconseillé de mettre ce paramètre à une valeur vide. En effet, cela peut provoquer une perturbation d'ActiveSync ou du Windows Mobile Device Center.*

#### 3.6.3. ExtendedInfo

Ce paramètre conditionne l'envoi des informations étendues lors d'un LIST. Les informations étendues (extended informations) sont les éléments suivants :

- Masque IP
- Passerelle (gateway)
- DHCP
- DNS
- WINS
- Numéro de série du constructeur du terminal. Les constructeurs suivants sont pris en compte:
  - Motorola (ESN),
  - Honeywell
  - PSION (EP10 seulement)
  - Intermec
- Les informations OEM étendues
- Les clés de registre spécifiées dans la section [RegKeys](#)
- Les valeurs (contenu de tag) XML spécifiées dans la section [XmlFiles](#)

#### Note

*Grâce à un mécanisme de plugin, il est possible de disposer du numéro de série pour n'importe quel constructeur. Il suffit pour cela de disposer d'une API C/C++ qui permet de retrouver ce numéro de série.*

Ces informations sont intégrées dans la base de données sous-jacente de Symphony.

Valeurs possibles	Signification
yes	Les informations étendues SONT incluses dans un LIST
no	Les informations étendues NE SONT PAS incluses dans un LIST

#### 3.6.4. ExtendedInfoOnUpdate

Ce paramètre conditionne l'envoi des informations étendues lors d'un UPDATE pour prise en compte lors de l'application des formules des collections.

#### Attention

*Ces informations ne sont pas prises en compte pour intégration dans la base de données sous-jacente de Symphony et elles ne sont donc pas visibles dans les UPDATES de l'historique d'une unité.*

#### Attention

*Ce paramètre est pris en compte seulement si le paramètre extendedinfo de la présente section est à yes.*

Les informations envoyées sont les suivantes:

- Masque IP si le paramètre extendedinfo de cette section est à yes
- Passerelle (gateway) si le paramètre extendedinfo de cette section est à yes
- DHCP si le paramètre extendedinfo de cette section est à yes
- WINS si le paramètre extendedinfo de cette section est à yes
- ESN Motorola ou serial sur PSION EP10 (vide si non disponible)
- Informations OEM si le paramètre oemInfo de cette section est à yes
- Valeurs des clés de registres spécifiées dans la section [RegKeys]
- Valeurs des tags XML spécifiés dans la section [XmlFiles]

Valeurs possibles	Signification
yes	Les informations étendues SONT incluses dans un UPDATE (voir signification exacte ci-dessus)
no	Les informations étendues NE SONT PAS incluses dans un UPDATE (voir signification exacte ci-dessus)

### 3.6.5. PhoneLineInfo

Ce paramètre conditionne l'envoi depuis le client des informations relatives au GPRS dans un LIST.

#### Attention

Ce paramètre est pris en compte seulement sur les PDAs sous Windows Mobile.

Les informations envoyées sont les suivantes:

- Statut ON/OFF du GPRS
- Statut réseau du GPRS
- Opérateur Télécom (fournisseur d'accès)
- Fabricant de la carte GPRS
- Version du firmware de la carte GPRS
- IMEI
- IMSI si disponible
- Type de WWAN (GSM, GPRS, EDGE, HSDPA, UMTS)
- Classe (GSMANDGPRS, GSMORGPRS, GSMORGPRS\_EXCLUSIVE, GPRSONLY, GSMONLY)
- Indicateur de présence d'un module radio/GPRS
- ICCID

Valeurs possibles	Signification
yes	Les informations GPRS SONT incluses dans un LIST
no	Les informations GPRS NE SONT PAS incluses dans un LIST

#### 3.6.6. OemInfo

Ce paramètre conditionne l'envoi des informations OEM dans un LIST (OEM = Original Equipment Manufacturer, c'est-à-dire le fabricant du PDA).

Les informations envoyées sont les suivantes:

- Version OEM
- Nom de l'OEM
- ID de la plateforme OEM
- ID de l'unité OEM

Valeurs possibles	Signification
yes	Les informations OEM SONT incluses dans un LIST
no	Les informations OEM NE SONT PAS incluses dans un LIST

#### 3.7. RegKeys



#### SECTION DYNAMIQUE

Configure un ensemble de valeurs de la base de registres Windows CE/Mobile.



#### Attention !

Cet ensemble de valeurs est transmis à la Console Symphony dans les commandes de type LIST si le paramètre **ExtendedInfo** est à yes.

Élément	Définition
Compteur	RegKeyCount
Lot	<ul style="list-style-type: none"><li>▪ RegKey<b>XX</b></li></ul>

#### 3.7.1. RegKeyCount

Compteur de la section dynamique.

La valeur maximale de ce compteur est 20. Si une valeur supérieure est fournie alors seules les 20 premières occurrences du lot de la section dynamique seront prises en compte.

Valeurs possibles	Signification
Valeur numérique entière	Renseigne le nombre total de clés de la base de registres configurées

#### 3.7.2. RegKeyXX

Indique le chemin d'une valeur dans la base de registre Windows CE/Mobile.

La syntaxe de ce paramètre est la suivante:

**regKeyXX**=[<chemin>]

où

- **XX** : l'index du lot courant sur 2 digits décimaux,
- **<chemin>** : le chemin de la valeur où la clé racine est représentée par une des abréviations spécifiées dans le tableau ci-après.

Valeurs possibles pour les clés racine	Définition	Exemple
HKLM	Abréviation pour la clé HKEY_LOCAL_MACHINE	regKey00=[HKLM\Drivers\Console\Font]
HKCU	Abréviation pour la clé HKEY_CURRENT_USER	regKey01=[HKCU\Drivers\Console\Font]
HKCR	Abréviation pour la clé HKEY_CLASSES_ROOT	regKey02=[HKCR\Drivers\Console\Font]
HKU	Abréviation pour la clé HKEY_USERS	regKey03=[HKU\Drivers\Console\Font]

### 3.8. XmlFiles



#### SECTION DYNAMIQUE

Configure un ensemble de chemin dans des fichiers XML.  
Les valeurs sont récupérées du fichier XML chaque fois que le serveur demande un LIST.

#### 3.8.1. XmlKeyCount

Compteurs de la section dynamique.

Valeurs possibles	Signification
Valeur numérique entière	Renseigne le nombre total de clés configurées dans des fichiers xml

#### 3.8.2. XmlXX

La syntaxe de ce paramètre est la suivante:

**XmlXX**=<Chemin>,<tag\_1>,...,<tag\_n>

où

- **XX** : l'index du lot courant de la section dynamique
- **<Chemin>** : Le chemin absolu du fichier XML à prendre en compte  
**<tag\_1>,...,<tag\_n>** : Le chemin du tag du fichier XML dont il faut remonter le contenu. Le premier tag représente la racine du document XML, et les tags suivants représentent l'arborescence à respecter:

```
<tag_1>
  . . .
  <tag_2>
    . . .
    <tag_n>
      contents
    </tag_n>
  . . .
</tag_2>
. . .
</tag_1>
```

S'il y a plusieurs possibilités dans le fichier XML, seule la première occurrence sera prise en compte. Si le tag <tag\_n> contient lui-même d'autres tags XML alors seul le texte immédiat sera pris en compte.



#### Exemple 1 :

Supposons qu'un fichier nommé `simple.xml` a le contenu suivant :

```
<breakfast_menu>
  <food>
    <price>12.00</price>
  </food>
</breakfast_menu>
```

et qu'il est dans le répertoire racine sur le terminal. Alors si la section `[XmlFiles]` est configurée comme suit:

```
[XmlFiles]
XmlKeyCount=1
Xml100=\simple.xml,<breakfast_menu>,<food>,<price>
```

la valeur "12.00" sera remontée.

#### Exemple 2 :

Supposons qu'un fichier nommé `simple.xml` a le contenu suivant :

```
<breakfast_menu>
  <food>
    <price>112.00<inner>113.00</inner>outer-100</price>
    <price>12.00<inner>13.00</inner>outer</price>
  </food>
</breakfast_menu>
```

et qu'il est dans le répertoire racine sur le terminal. Alors si la section `[XmlFiles]` est configurée comme suit:

```
[XmlFiles]
XmlKeyCount=1
Xml100=\simple.xml,<breakfast_menu>,<food>,<price>
```

la valeur "112.00" sera remontée.

## 3.9. Stats

Paramétrage des options de traitement du monitoring et statistiques (module WdpStats).

### 3.9.1. CipherStats

Active/désactive le cryptage des fichiers de statistiques envoyés.

Valeurs possibles	Signification
yes	Active le cryptage des fichiers de statistiques envoyés
no	Désactive le cryptage des fichiers de statistiques envoyés

### 3.9.2. CipherKey

Définit la clé de cryptage de 64 digits hexadécimaux pour encrypter les fichiers de statistiques transmis à la console. Le cryptage utilisé est l'AES 256 CBC.

Valeurs possibles	Signification
Clé de 64 digits hexadécimaux	Clé de cryptage sur 256 bits utilisée pour le chiffrement des fichiers de statistiques transmis à la console

**Exemple :**

```
cipherKey=6162636465666768696A6B6C6D6E6F704142434445464748494A4B4C4D4E4F50
```

### 3.9.3. SendStats

Active/désactive l'envoi des fichiers de statistiques vers le serveur.

Valeurs possibles	Signification
yes	Active l'envoi des fichiers de statistiques vers le serveur
no	Désactive l'envoi des fichiers de statistiques vers le serveur

### 3.9.4. SendVxtLog

Envoie les fichiers journaux de voiXtreme sur le serveur.

Valeurs possibles	Signification
yes	Active l'envoi des fichiers journaux de voiXtreme vers le serveur
no	Désactive l'envoi des fichiers journaux de voiXtreme vers le serveur

### 3.10. Time

Paramétrage de synchronisation de l'horloge interne du terminal.

#### 3.10.1. SyncTimeWdp

Synchronise l'heure locale avec celle du serveur qui héberge la Console Symphony.  
Configurer un serveur comme « serveur principal de temps » (Main Time Server).

Valeurs possibles	Signification
yes	Active la synchronisation de l'heure locale avec celle du serveur
no	Désactive la synchronisation de l'heure locale avec celle du serveur

#### 3.10.2. SyncTimeNtp

Synchronise l'heure avec celle du serveur public NTP (voir [serverNtp](#)).

Valeurs possibles	Signification
yes	Active la synchronisation de l'heure avec celle du serveur public NTP
no	Désactive la synchronisation de l'heure avec celle du serveur public NTP

#### 3.10.3. TimeZone

Renseigne en temps (heures) le décalage avec l'heure UMT. Ce nombre d'heure sera retranché de l'heure système (UMT).

Valeurs possibles	Signification
Valeur numérique entière	Définit le nombre d'heures de décalage avec UMT La valeur numérique 99 signifie que cette option est désactivée (garde la timezone du système)

#### 3.10.4. ServerNtp

Renseigne une liste de serveur de temps pour synchroniser par NTP (noms séparés par des virgules).

Valeurs possibles	Signification
Chaîne de caractères	Liste les serveurs de temps utilisés, séparés par une virgule

#### Exemple :

```
serverNtp=time.windows.com,time.nist.gov,nist1-la.ustiming.org,time-nw.nist.gov
```

#### 3.10.5. PeriodNtp

Renseigne le temps en minutes entre chaque tentative de connexion NTP.

Valeurs possibles	Signification
Valeur	Définit l'intervalle de temps en minutes entre chaque connexion NTP

numérique  
entière

### 3.11. AutoUpdate

Cette section permet de contrôler les commandes effectuées de façon automatique par le client Symphony. Ces commandes automatiques peuvent être:

- Transmettre ses informations et son statut (LIST),
- Demander une mise à jour des paquets (UPDATE),
- Faire un LIST **ET** un UPDATE

Ces commandes automatiques peuvent effectuées/planifiées selon les possibilités suivantes:

- Commande(s) effectuée(s) au démarrage de l'application (paramètres [DoStartupCommand](#) et [StartupCommand](#))
- Commande(s) effectuée(s) de façon périodique ([DoPeriodic](#) et [PeriodicCommand](#))
- Commande(s) effectuée(s) quotidiennement à une certaine heure ([DoSchedule](#) et [ScheduleCommand](#))
- Commande(s) effectuée(s) à la détection de l'établissement d'une connexion réseau ([OnConnect](#), [OnConnectInterval](#) et [OnConnectCommand](#))

Le client Symphony possède aussi une fonctionnalité de **gestion des plages d'activité** pour les commandes périodiques. Par défaut, cette fonctionnalité est en "tout ou rien": soit elle n'est pas activée auquel cas le client transmet en permanence sa commande périodique. Dans certains cas, il peut être utile voire bénéfique de limiter le fonctionnement de cette commande périodique à des plages de temps données. En effet, chaque commande (LIST ou UPDATE) donne lieu à une intégration en base de données (pour pouvoir être "historisée") et consomme donc de la ressource (CPU et espace disque). Les 2 paramètres [doWeekActivity](#) et [weekActivity](#) permettent de contrôler cette fonctionnalité.

#### 3.11.1. DoStartupCommand

Active/désactive l'exécution d'une commande au démarrage de l'application. La commande à exécuter est spécifiée dans le paramètre [startUpcommand](#).

Valeurs possibles	Signification
yes	Active la startUpcommand
no	Désactive la startUpcommand

#### 3.11.2. StartupCommand

Renseigne les commandes à exécuter lors du lancement du client.

Valeurs possibles	Signification
L	Exécute la commande LIST (Rapporter)
U	Exécute la commande UPDATE (Mise à jour paquets)
LU	Exécute les commandes LIST (Rapporter) et UPDATE (Mise à jour paquets)

#### 3.11.3. StartupCommandTimeout

Si aucune connexion réseau n'est disponible au démarrage de l'application, SpnClient va se mettre en attente d'une connexion valide avant de lancer la commande spécifiée par [StartupCommand](#). Ce paramètre fixe le délai maximum pour cette attente (timeout) en secondes.

Valeurs possibles	Signification
Valeur numérique	Définit le temps en secondes pendant lequel startUpCommand sera exécuté

entière

#### 3.11.4. DoPeriodic

Active/désactive les commandes périodiques. Les commandes périodiques sont effectuées à intervalle régulier.

Valeurs possibles	Signification
yes	Active les connexions périodiques
no	Désactive les connexions périodiques

#### 3.11.5. Period

Période en minutes pour effectuer les tâches périodiques.

Valeurs possibles	Signification
Valeur numérique entière	Définit l'intervalle de temps en <b>minutes</b> entre deux connexions périodiques consécutives

#### 3.11.6. PeriodicCommand

Commande à exécuter lors des commandes périodiques.

Valeurs possibles	Signification
L	Exécute la commande LIST (Rapporter)
U	Exécute la commande UPDATE (Mise à jour paquets)
LU	Exécute les commandes LIST (Rapporter) et UPDATE (Mise à jour paquets)

#### 3.11.7. PeriodicWakeUpDevice

Réveille le terminal afin de lancer la commande configurée dans le paramètre ***periodicCommand***. Ce paramètre sert à s'assurer qu'un terminal en veille sera réveillé pour effectuer une comande périodique.

Valeurs possibles	Signification
yes	Active le terminal pour lancer la commande configurée dans periodicCommand
no	Désactive l'option d'activation du terminal pour la commande configurée dans periodicCommand

#### 3.11.8. DoSchedule

Active/désactive les commandes programmées.

Valeurs possibles	Signification
yes	Active les connexions programmées
no	Désactive les connexions programmées

### 3.11.9. ScheduleCommand

Commande à exécuter lors des tâches programmées.

Valeurs possibles	Signification
L	Exécute la commande LIST (Rapporter)
U	Exécute la commande UPDATE (Mise à jour paquets)
LU	Exécute les commandes LIST (Rapporter) et UPDATE (Mise à jour paquets)

### 3.11.10. OnConnect

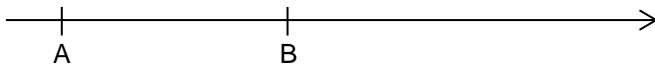
Active/désactive l'exécution d'une commande à la détection de l'établissement de la connexion réseau.

Valeurs possibles	Signification
yes	Envoie une commande à la détection de la connexion réseau
no	Désactive l'envoi d'une commande à la détection de la connexion réseau

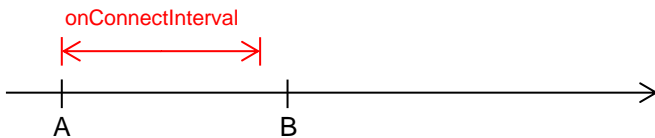
### 3.11.11. OnConnectInterval

Ce paramètre spécifie le temps minimum en secondes entre 2 événements de connexions pour que la commande soit prise en compte.

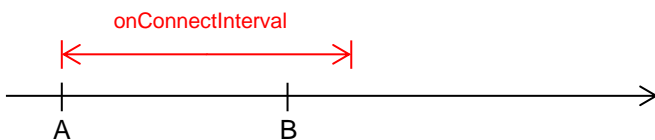
En effet, supposons que 2 connexions réseau soient détectées et nommons A (la première) et B (la deuxième) ces 2 événements sur un axe temporel :



Si l'écart de temps entre A et B est plus **grand** que onConnectInterval alors la commande qui est causée par la connexion détectée en B **est prise en compte**. Voici une représentation graphique de cela sur l'axe temporel:



Si l'écart de temps entre A et B est plus **petit** que onConnectInterval alors la commande qui est causée par la connexion détectée en B n'est **pas** prise en compte. Voici une représentation graphique de cela sur l'axe temporel:



Valeurs possibles	Signification
Valeur numérique entière	Définit un temps d'intervalle en secondes

### 3.11.12. OnConnectCommand

Commande à exécuter lorsqu'une connexion réseau est détectée.

Valeurs possibles	Signification
L	Exécute la commande LIST (Rapporter)
U	Exécute la commande UPDATE (Mise à jour paquets)
LU	Exécute les commandes LIST (Rapporter) et UPDATE (Mise à jour paquets)

### 3.11.13. doWeekActivity

Active ou désactive la gestion de plages d'activité pour les commandes périodiques. Cette gestion se fait sur la base d'un emploi du temps hebdomadaire spécifié par le paramètre weekActivity de la présente section.

Valeurs possibles	Signification
yes	Active les connexions programmées
no	Désactive les connexions programmées

Quand ce paramètre est à **no** et que les connexions périodiques sont activées (doPeriodic=yes), le client Symphony exécute **EN PERMANENCE** sa commande périodique.

Quand ce paramètre est à **yes** et que les connexions périodiques sont activées (doPeriodic=yes), le client Symphony exécute sa commande périodique **UNIQUEMENT PENDANT LES PLAGES DE TEMPS** spécifiées dans le paramètre weekActivity.

### 3.11.14. weekActivity

Ce paramètre définit les plages d'activités pour la commande périodique du client Symphony. Ces plages d'activité sont définies sur la base d'un emploi du temps hebdomadaire: chaque jour de la semaine est affecté de 1 à 6 plage(s) horaire(s). Plus précisément la syntaxe à la forme suivante:

La syntaxe est la suivante (les symboles fixes de la syntaxe sont en rouge et en gras, par exemple: **LUN**) :

**weekActivity=DDD<sub>1</sub>(Time\_slices<sub>1</sub>) DDD<sub>2</sub>(Time\_slices<sub>2</sub>) ... DDD<sub>n</sub>(Time\_slices<sub>n</sub>)**

où :

- **DDD<sub>i</sub> ::= LUN|MAR|MER|JEU|VEN|SAM|DIM|SUN|MON|TUE|WED|THU|FRI|SAT**  
**DDD<sub>i</sub>** code le jour de la semaine avec la signification suivante :

Valeurs	Jour spécifié
<b>LUN</b>	Lundi
<b>MAR</b>	Mardi
<b>MER</b>	Mercredi
<b>JEU</b>	Jeudi
<b>VEN</b>	Vendredi
<b>SAM</b>	Samedi
<b>DIM</b>	Dimanche
<b>SUN</b>	Dimanche
<b>MON</b>	Lundi
<b>TUE</b>	Mardi
<b>WED</b>	Mercredi



<b>THU</b>	Jeudi
<b>FRI</b>	Vendredi
<b>SAT</b>	Samedi

Chaque jour de la semaine peut être présent au plus une fois sinon une erreur de syntaxe est détectée. Si un jour de la semaine n'est pas spécifié alors le client Symphony n'exécutera pas sa commande périodique entre 00h00 et 23h59 pour le jour concerné.

- Chaque élément *Time\_slices*, à la forme *time\_slice*(*time\_slice*)+ où
  - *time\_slice* ::= [0-9][0-9]:[0-9][0-9]-[0-9][0-9]:[0-9][0-9]*time\_slice* code une plage de temps. La première heure est l'heure de début de la plage de temps et la deuxième (celle après le tiret -) est l'heure de fin de la plage de temps.  
Exemple: 08:30-18:25 code la plage de temps entre 8h30 et 18h25 pour le jour en cours.

Il peut y avoir au plus 6 éléments de type *Time\_slice* (séparées par le caractère pipe |) pour un jour donné. Exemple : 08:30-12:30|13:00-17:00 est un élément de type *Time\_slices* qui code les plages de temps de 8h30 à 12h30 puis de 13h00 à 17h00.

Les heures doivent être comprises entre 00 et 23 et les minutes entre 00 et 59. Si un de ces 2 éléments n'est pas dans son intervalle autorisé alors il y aura une erreur de syntaxe.

Si une erreur de syntaxe est rencontrée dans le paramètre *WeekActivity* avec le paramètre *doWeekActivity* à yes alors le client Symphony sera actif de façon permanente au niveau des commandes périodiques. C'est à dire qu'il aura la plage d'activité 00:00-23:59 pour tous les jours de la semaine.

Voici ci-dessous un exemple complet pour le paramètre *weekActivity*.

#### Exemple 1 :

On reproduit ci-dessous la section [autoUpdate] au complet :

```
[autoUpdate]
doStartupCommand=yes
startupCommand=LU
startupCommandTimeout=60
doPeriodic=yes
period=2
periodicCommand=LU
periodicWakeUpDevice=yes
doSchedule=yes
scheduleCommand=L
onConnect=no
onConnectInterval=600
onConnectCommand=LU
doWeekActivity=yes
WeekActivity=LUN(10:20-10:30|10:40-10:50|11:00-11:10|11:20-11:30) MAR(16:50-17:00) JEU(06:30-10:00|17:50-17:55) FRI(08:00-19:00)
```

Dans l'exemple ci-dessus, la fonctionnalité de gestion des plages de temps est activée et le client Symphony exécute sa commande périodique (LIST et UPDATE) selon l'emploi du temps hebdomadaire suivant :

- Lundi: de 10h20 à 10h30, de 10h40 à 10h50, de 11h00 à 11h10 et de 11h20 à 11h30
- Mardi: de 16h50 à 17h00
- Jeudi: de 06h30 à 10h00 et de 17h50 à 17h55
- Vendredi: de 08h00 à 19h00

En dehors de ces plages de temps, le client Symphony n'exécute pas sa commande périodique.

### 3.12. Security

Cette section permet de contrôler le comportement du client Symphony pour la fonctionnalité de détection de non réponse du serveur pendant les commandes périodiques. On parlera de la fonctionnalité de "Dead server monitoring". Plus précisément, il s'agit de fixer une durée en minutes (timeout) qui sera utilisée comme suit: si le serveur n'a pas répondu aux demandes du terminal lors de ses connexions périodiques sur une durée excédant ce timeout alors un script .wdp est exécuté.

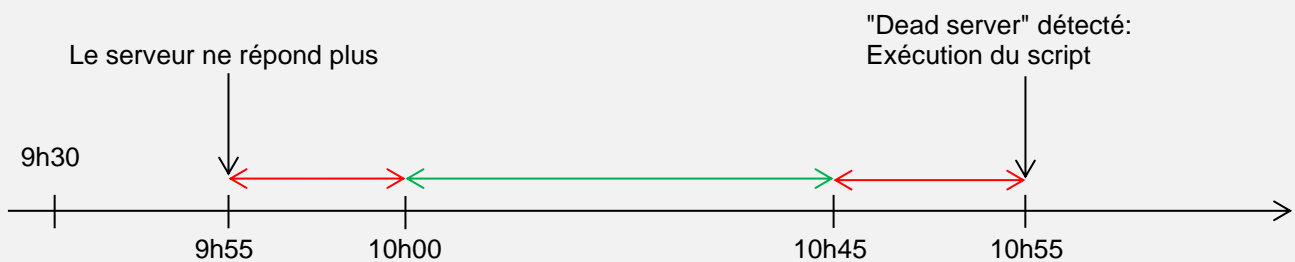
Cette fonctionnalité est liée à celle de gestion des plages d'activité des commandes périodiques (détaillée dans les paragraphes [3.11.13](#) et [3.11.14](#)). En effet, seul le temps pendant lequel le client est dans une plage d'activité **ET** pendant lequel le serveur ne répond pas est comptabilisé pour le chronomètre interne au client Symphony. Cela signifie en d'autres termes que les plages de temps où le client Symphony n'est pas activé pour les commandes périodiques ne sont pas prises en compte. Voici un exemple complet qui illustre l'interaction entre le "Dead server monitoring" et les plages d'activités des commandes périodiques.

#### Exemple :

Supposons que l'on ait le paramétrage suivant :

- La commande périodique est activée avec une période de 5 minutes.
- La gestion des plages d'activité est activée et elle est paramétrée sur les tranches horaires 08:00-10:00 | 10:45-18:00 pour le jour en cours.
- Le paramètre `deadServerTimeout` a la valeur 15.

Supposons que le terminal démarre à 9h30 et qu'à partir de 9h55 le serveur ne répond plus. Alors la détection de "Dead server" n'aura lieu qu'à 10h55 car le temps entre 10h00 et 10h45 n'est pas comptabilisé. Seules les durées entre 9h55 et 10h00 (5 minutes) et celle entre 10h45 et 10h55 (10 minutes) ont été prises en compte. En effet la tranche de temps entre 10h00 et 10h45 est une plage d'inactivité.



#### 3.12.1. doMonitorDeadServer

Active/désactive la fonctionnalité de "Dead server monitoring".

Valeurs possibles	Signification
yes	Active la fonctionnalité de "Dead server monitoring"
no	Désactive la fonctionnalité de "Dead server monitoring"

#### 3.12.2. deadServerTimeout

Durée en minutes du timeout à atteindre sans réponse du serveur avant d'exécuter le script .wdp spécifié dans le paramètre deadServerScript.

Valeurs possibles	Signification
Valeur numérique entière	Durée en minutes au-delà de laquelle est détectée un événement de type "Dead server".

#### 3.12.3. deadServerScript

Valeurs possibles	Signification
Chaînes de caractères	Chemin du script .wdp (sur le terminal) qui sera exécuté en cas de Détection de "Dead server".

#### 3.12.4. repeatDeadServerScript

Active/désactive la répétition du script spécifié dans le paramètre deadServerScript.

Valeurs possibles	Signification
yes	Le script .wdp à appeler en cas de détection de "Dead server" <b>EST</b> répété.
no	Le script .wdp à appeler en cas de détection de "Dead server" <b>N'EST PAS</b> répété.

### 3.13. Schedule



#### SECTION DYNAMIQUE

Le client Symphony Windows CE possède une fonctionnalité de connexion programmée : schedule connect. Cette fonctionnalité consiste à programmer une connexion (un LIST, un UPDATE ou les 2) dans une plage de temps donnée avec un délai pour recommencer périodiquement tant qu'il y a eu un échec et qu'on reste dans la plage de temps (entre l'heure spécifiée par **ConnStartXX** et celle spécifiée par **ConnEndXX**).

Les tentatives de connexions sur un schedule connect s'arrêtent à la première connexion effectuée avec succès ou si on est sorti de la plage de temps. Une connexion de type schedule connect permet aussi de venir réveiller le terminal au besoin (si **ConnWakeUpDeviceXX** est à yes).

Élément	Définition
Compteur	ConnCount
Lot	<ul style="list-style-type: none"> <li>▪ ConnStartXX</li> <li>▪ ConnEndXX</li> <li>▪ ConnPeriodXX</li> <li>▪ ConnWakeUpDeviceXX</li> </ul>

Chaque occurrence d'un lot constitue une connexion programmée.



#### Note

Toutes les connexions programmées effectuent la même commande : celle qui est spécifiée dans l'option **ScheduleCommand**



#### Attention !

Pour que les éléments de cette section soient pris en compte, il est nécessaire que le paramètre **doSchedule** de la section **AutoUpdate** soit à yes

#### 3.13.1. ConnCount

Compteur de la section dynamique.

Valeurs possibles	Signification
Valeur numérique	Renseigne le nombre total de connexions programmées

#### 3.13.2. ConnStartXX

Spécifie l'heure de début pour lancer les tentatives de connexion.

Valeurs possibles	Signification	Syntaxe dans fichier de configuration
Valeur numérique	Indique l'heure de début au format HH:MM où la tentative de connexion sera tentée	connStart00=14:37

#### 3.13.3. ConnEndXX

Spécifie l'heure de fin pour les tentatives de connexion.

##### Note

*Même si aucune tentative de connexion n'a abouti pour une connexion programmée, l'application stoppe la connexion programmée au-delà de l'heure de fin spécifiée par le présent paramètre.*

Valeurs possibles	Signification	Syntaxe dans fichier de configuration
Valeur numérique	Indique l'heure de fin au format HH:MM où la tentative de connexion sera arrêtée	connEnd00=15:00

#### 3.13.4. ConnPeriodXX

Définit la durée de la période pour les tentatives de connexions périodiques qui ont lieu en cas d'échec.

##### Note

*Dès qu'une connexion programmée a réussi, plus aucune tentative n'est faite sur la connexion programmée en question.*

Valeurs possibles	Signification
Valeur numérique	Indique le temps de pause entre chaque tentative en minutes (période)

#### 3.13.5. ConnWakeUpDeviceXX

Active/désactive la sortie automatique de veille pour réaliser la connexion programmée.

Valeurs possibles	Signification
yes	Active l'option
no	Désactive l'option

### 3.14. ScheduleScript



#### SECTION DYNAMIQUE

Le client Symphony peut exécuter automatiquement des actions contenues dans un script .wdp de façon programmée. Cette fonctionnalité se nomme les *Scheduled Scripts*. L'exécution automatique a lieu quotidiennement à une heure fixe.

Élément	Définition
Compteur	Count
Lot	<ul style="list-style-type: none"> <li>▪ Time<code>XX</code></li> <li>▪ Script<code>XX</code></li> </ul>

Chaque lot au sein de la section dynamique définit une action programmée (scheduled script)

#### Note

Les *Scheduled Scripts* n'affichent pas d'interface graphique ni de messages à l'utilisateur comme peuvent le faire les scripts .wdp exécutés au démarrage (voir la section [4.1 Exécution automatique de scripts](#))

#### 3.14.1. Count

Comptabilise le nombre total de scripts wdp programmés.

Valeurs possibles	Signification
Valeur numérique entière	Renseigne le nombre total de scripts wdp programmés

#### 3.14.2. Time`XX`

Définit l'heure d'exécution du script wdp au format HH:MM

Valeurs possibles	Signification	Syntaxe dans fichier de configuration
Valeur numérique	Indique l'heure au format HH :MM d'exécution du script	time00=16:23

#### 3.14.3. Script`XX`

Indique le chemin absolu du script .wdp qui doit être exécuté.

Valeurs possibles	Signification
Chaîne de caractères	Indique le chemin complet sur le terminal pour accéder au script wdp

### 3.15. Deployment

Cette section permet de configurer les paramètres de déploiement de paquets.

#### 3.15.1. SkipOnError

Le processus UPDATE ignore les erreurs au téléchargement et continue avec les paquets suivants.

Valeurs possibles	Signification
yes	Active l'option pour ignorer les erreurs au téléchargement
no	Désactive l'option

#### 3.15.2. SendUpdateLog

Envoie au serveur la dernière mise à jour du journal (log) généré.

Valeurs possibles	Signification
yes	Active l'option pour envoyer la dernière mise à jour de log
no	Désactive l'option

#### 3.15.3. SilentDeployment

Le processus UPDATE se lance à l'arrière-plan.  
Aucune fenêtre ne s'affiche.

Valeurs possibles	Signification
yes	Active l'affichage d'une interface utilisateur lors d'un update
no	Désactive l'option

#### 3.15.4. AskToInstall

Autorise l'utilisateur à accepter ou refuser l'installation des paquets.  
Si ce paramètre est à yes alors une fenêtre s'affichera lors de tout déploiement de paquet et demandera confirmation à l'utilisateur. Si l'utilisateur ne répond pas au-delà d'un certain temps, SpnClient considère que l'utilisateur n'a pas confirmé et n'installera pas le paquet en question.

Valeurs possibles	Signification
yes	Active l'option pour autoriser l'installation des paquets
no	Désactive l'option

#### 3.15.5. AskToInstallNo

Temps pour répondre en secondes à une demande de confirmation d'installation de paquets (voir ci-dessus).

Valeurs possibles	Signification
Valeurs numériques entières	Active l'option pour cacher le processus d'update des paquets

#### 3.15.6. WarmBootExe

Spécifie le chemin absolu d'un exécutable à lancer lors d'une commande de warmboot.

Quand le client Symphony doit effectuer un warmboot (dans un script .wdp, dans un paquet ou par le biais de la Console Symphony), ce dernier contrôle si ce paramètre est renseigné (non vide) et si c'est le cas, il utilise l'exécutable spécifié. Dans le cas contraire (si ce paramètre est vide), le client Symphony utilise les API Win32 usuelles pour effectuer le warmboot.

Valeurs possibles	Signification
Chaine de caractères	Donne le nom du programme alternatif à exécuter en cas de commande de Warm Boot.

#### 3.15.7. ColdBootExe

Ce paramètre a la même fonction que le paramètre [WarmBootExe](#) ci-dessus excepté qu'il concerne le **coldboot**.

Valeurs possibles	Signification
Chaine de caractères	Donne le nom du programme alternatif à exécuter en cas de commande de Cold Boot.



## 4. Gestion du démarrage

### 4.1. Exécution automatique de scripts

Au démarrage de l'application, le client Symphony peut exécuter un ou plusieurs scripts qui contiennent des actions. Ces scripts doivent respecter la syntaxe décrite dans le paragraphe [5. Syntaxe des scripts wdp](#).

En effet, quand le client Symphony démarre, ce dernier recherche les fichiers dont le nom respecte la syntaxe \*.wdp\* dans le répertoire **parent** du répertoire wdp (voir [6.3 Répertoire d'installation](#)) qui contient l'exécutable du client Symphony (SpnClient.exe).

Afin de garantir un ordre dans l'exécution séquentielle de plusieurs scripts, les fichiers sont traités dans l'ordre suivants : .wdp, .wdp0, .wdp1, .wdp2, ... , .wdp5.



#### Attention !

*Les scripts dont l'extension est wdp, wdp0 et wdp1 seront traités **uniquement en cas de Cold Boot**.*



#### Attention !

*Si un script qui a l'extension wdp5 est trouvé, il sera exécuté puis le client Symphony se **fermera**.*

## 5. Syntaxe des scripts wdp

Un script wdp est un fichier au format Microsoft INI comme l'est le fichier de configuration du client Symphony : SpnClient.cfg. Comme ce dernier, un script wdp peut comporter des sections dynamiques. Néanmoins, certaines sections dites dynamiques pourront comporter d'autres éléments fixes comme l'est le compteur.

### Note

*Une section dynamique sera indiquée et spécifiée de la même façon que pour le fichier de configuration comme expliqué dans le paragraphe [2.5 Sections dynamiques](#).*

Dans la plupart des actions possibles pour un script wdp, il est demandé de spécifier des chemins de fichiers/répertoires. Tous les chemins spécifiés peuvent comporter des variables d'environnement qui sont automatiquement remplacées par leur valeur à l'exécution. Voici un tableau récapitulatif de ces variables :

Variable	Contenu
%PDA_DESKTOP%	Chemin du dossier qui contient physiquement les éléments du bureau Windows CE/Mobile
%PDA_PROGRAMS%	Chemin du dossier qui contient les groupes de programmes de l'utilisateur (qui sont eux-mêmes des répertoires)
%PDA_PERSONAL%	Chemin du dossier commun pour stocker les documents
%PDA_STARTUP%	Chemin du dossier à lancer au démarrage. Tous les éléments de ce dossier sont lancés automatiquement au démarrage du terminal.
%PDA_RECENT%	Chemin du dossier contenant les documents récemment ouverts par l'utilisateur.
%PDA_FONTS%	Répertoire virtuel contenant les polices de caractères.
%PDA_FAVORITES%	Chemin du dossier contenant les favoris de l'utilisateur.
%PDA_WINDOWS%	Chemin du dossier Windows
%PDA_PROGRAM_FILES%	Chemin du dossier où sont installés les programmes
%PDA_START_MENU%	Chemin du dossier qui contient les éléments du menu Démarrer.

Chacune des sections possibles est décrite dans les paragraphes ci-dessous (6.x) et chaque sous-paragraphe en décrit les valeurs (numérotés 6.x.y).

**Exemple pour un fichier HelpV100.wdp**

```
[ACTION-ZIP]
FileCount=1
File00=\Application\Help.zip

[ACTION-CPY]
FileCount=2
File00=\Application\Help\HelpViewer.exe > \Windows\Help\HelpViewer.exe
File01=\Application\Help\HelpVw.lnk > \Windows\Help\HelpVw.lnk

[ACTION-REG]
FileCount=1
File00=\Application\Help\Config.reg

[ACTION-DEL]
FileCount=3
File00=\Application\Help\Config.reg
File01=\Application\Help.zip
File02=\Application\HelpV100.wdp
```

**5.1. MAIN**

Cette section permet de paramétrer l'exécution du script de façon générale. Si elle est présente, elle doit l'être en un seul exemple.

**5.1.1. delayBeforeRunning**

Cette option permet d'ajouter une temporisation exprimée en **millisecondes** avant l'exécution de toute action du script.

Valeurs possibles	Définition	Exemple
Entier positif	Nombre de millisecondes à attendre avant d'exécuter les actions du script.	delayBeforeRunning=30000

**5.1.2. sequentialActions**

Cette option permet de paramétrer le séquençement des actions présentes dans le script. Le séquençement des actions du script a 2 modes d'exécution (mutuellement exclusifs):

- Mode d'exécution **séquentiel** :  
Les actions sont exécutées dans l'ordre où elles se trouvent. Il est possible de dupliquer une action (et donc d'avoir plusieurs fois la même section)
- Mode d'exécution **non séquentiel** :  
Les actions sont exécutées dans un certain ordre (voir ci-dessous). Les actions ne peuvent pas être dupliquées. Si une action est dupliquée, seule la première occurrence sera prise en compte. Dans ce mode d'exécution, les actions sont effectuées dans l'ordre suivant:
  - ACTION-ZIP
  - ACTION-CPY
  - ACTION-ATTRIB
  - ACTION-REG
  - ACTION-WAIT-EXE
  - ACTION-EXE
  - ACTION-REGEXP
  - ACTION-DEL
  - ACTION-REMOVE
  - ACTION-KILL
  - ACTION-DELAY

Valeurs possibles	Définition	Syntaxe dans fichier de configuration
yes   no	Active/désactive le mode d'exécution séquentiel des actions du script	<code>sequentialActions=yes</code>

**Note**

Par défaut, le mode d'exécution utilisé est le mode **non séquentiel**.

**Exemple :** Il est demandé une temporisation de 20 secondes, le script sera effacé automatiquement après exécution et les actions seront effectuées selon le mode séquentiel (voir explications ci-dessous).

```
[MAIN]
delayBeforeRunning=20000
autodelete=yes
sequentialActions=yes
```

### 5.1.3. autodelete

Cette option permet d'effacer automatiquement le script wdp à la fin de son exécution.

Valeurs possibles	Définition	Syntaxe dans fichier de configuration
yes	Le script <b>est automatiquement effacé</b> à la fin de son exécution.	<code>autodelete=yes</code>
no	Le script <b>n'est pas automatiquement effacé</b> à la fin de son exécution.	<code>autodelete=no</code>

**Note**

Par défaut, un script n'est pas effacé en fin d'exécution.

## 5.2. ACTION-ZIP

 SECTION DYNAMIQUE

Cette action permet de décompresser une archive de type ZIP sur le terminal.

Élément	Définition
Compteur	FileCount
Éléments fixes	<ul style="list-style-type: none"> <li>stoponerror</li> </ul>
Lot	<ul style="list-style-type: none"> <li>File<b>XX</b></li> </ul>

**Exemple :**

```
[action-zip]
filecount =2
file00 = \flash\application.zip
file01 = \configuration.zip
```

5.2.1. FileCount

Il s'agit du compteur de la section dynamique qui indique le nombre de fichiers à décompresser pour cette action.

5.2.2. Stoponerror

Cette option permet de terminer l'action à la première erreur rencontrée dans le cas où plusieurs fichiers sont à traiter.

Valeurs possibles	Définition	Syntaxe dans fichier de configuration
yes	L'action s'arrête à la première erreur rencontrée.	stoponerror=yes
no	L'action continue son exécution si une erreur est recontrée	stoponerror=no

5.2.3. FileXX

Chemin absolu du fichier à décompresser. Les chemins qui comportent des espaces ne doivent pas nécessairement être entourés de guillemets.

**5.3. ACTION-CPY**

Cette action permet d'effectuer des copies de fichiers sur le terminal.



**SECTION DYNAMIQUE**

Elément	Définition
Compteur	FileCount
Éléments fixes	<ul style="list-style-type: none"> <li>▪ stoponerror</li> </ul>
Lot	<ul style="list-style-type: none"> <li>▪ FileXX</li> </ul>

**Exemple**

```
[ACTION-CPY]
FileCount=1
File00=\Flash Disk\Restore.cpy
```

5.3.1. FileCount

Il s'agit du compteur de la section dynamique qui indique le nombre de copies de fichier à effectuer pour cette action.

### 5.3.2. stoponerror

Idem que pour la section précédente.

### 5.3.3. FileXX

Spécifie un fichier CPY qui contient les instructions de copie. Il s'agit d'un fichier de type texte en encodage ASCII (ANSI) où chaque ligne doit avoir la syntaxe suivante :

```
source > destination
```

où :

- `source` : Chemin absolu du fichier source sur le terminal
- `destination` : Chemin absolu du fichier de destination sur le terminal

### Exemple

```
\wdp\flexvnc.exe > \Windows\flexvnc.exe
\wdp\flexftpserver.exe > \Windows\flexftpserver.exe
\wdp\flexrpsvserver.exe > \Windows\flexrpsvserver.exe
\flash\myapp.lnk > \program files\myapp.lnk
\wdp\flexvnc.exe > %PDA_WINDOWS%\flexvnc.exe
\wdp\flexftpserver.exe > %PDA_WINDOWS%\flexftpserver.exe
\wdp\flexrpsvserver.exe > %PDA_WINDOWS%\flexrpsvserver.exe
\flash\myapp.lnk > %PDA_PROGRAM_FILES%\myapp.lnk
%PDA_PROGRAM_FILES%\a.lnk > %PDA_PROGRAMS%\a.lnk
```

### Note

Si un chemin (source ou cible) comporte des espaces (par exemple `\Program Files`), il n'est pas nécessaire de l'entourer de guillemets dans un fichier CPY.

Il est possible d'utiliser les variables d'environnement listées au début du paragraphe [5 Syntaxe des scripts wdp](#).

## 5.4. ACTION-ATTRIB

Cette action permet de modifier les attributs d'un fichier sur le terminal (fichier caché et/ou en lecture seule).



### SECTION DYNAMIQUE

Élément	Définition
Compteur	FileCount
Éléments fixes	<ul style="list-style-type: none"><li>▪ stoponerror</li></ul>
Lot	<ul style="list-style-type: none"><li>▪ FileXX</li></ul>

### 5.4.1. FileCount

Il s'agit du compteur de la section dynamique qui indique le nombre de fichiers à traiter pour cette action.

### 5.4.2. stoponerror

Idem que pour la section précédente.

## 5.4.3. FileXX

Cet élément spécifie le fichier à traiter avec les attributs à appliquer avec la syntaxe suivante:

```
[File name][SPACE][attributes]
```

où

- [File name] :Chemin absolu du fichier à traiter. Il doit être entouré de guillemets (double quotes ") s'il comporte un ou plusieurs espaces.
- [SPACE] : Le caractère espace.
- [attributes] : Le ou les attributs (séparés ou non par un espace) à appliquer pour le fichier:
  - +h : ajouter l'attribut de fichier caché
  - -h : supprimer l'attribut de fichier caché
  - +r : ajouter l'attribut de lecture seule
  - -r : supprimer l'attribut de lecture seule
- Pour des raisons de cohérence, il n'est pas possible de supprimer et d'ajouter un attribut en même temps et cela sera considéré comme une erreur.
- 

**Exemple:** Le fichier n'est plus caché et il est en lecture seule.

```
[ACTION-ATTRIB]
FileCount=1
File00=\Flash Disk\WTnCe.exe -h +r
```

## 5.5. ACTION-REG

Cette action permet de mettre à jour la base de registre du terminal en appliquant un fichier de type .reg au format REGEDIT4. Un mot clef %DELETE% permet d'effacer une clef ou une valeur :

- Pour effacer une **valeur**, il faut mettre ce mot clé **directement après la clé** concernée dans le fichier .reg.
- Pour effacer une **clé**, il faut lui **attribuer la valeur** %DELETE% dans le fichier .reg.

**Exemple:**

```
REGEDIT4
[HKEY_CURRENT_USER\Cz\test]
"AllowChange"=dword:00000001
[HKEY_CURRENT_USER\Cz\test]
"AllowChange"=%DELETE%
[HKEY_CURRENT_USER\Cz\test]
%DELETE%
```



## SECTION DYNAMIQUE

Élément	Définition
Compteur	FileCount
Éléments fixes	<ul style="list-style-type: none"> <li>▪ stoponerror</li> </ul>
Lot	<ul style="list-style-type: none"> <li>▪ File<b>XX</b></li> </ul>

### 5.5.1. FileCount

Il s'agit du compteur de la section dynamique qui indique le nombre de fichiers .reg à traiter pour cette action.

### 5.5.2. stoponerror

Idem que pour la section précédente.

### 5.5.3. FileXX

Chemin absolu du fichier .reg au format REGEDIT 4 à appliquer.

#### Note

*Si le chemin comporte des espaces (par exemple \Program Files), il n'est pas nécessaire de l'entourer de guillemets.*

*Il est possible d'utiliser les variables d'environnement listées au début du paragraphe [5 Syntaxe des scripts wdp](#).*

## 5.6. ACTION-WAIT-EXE

Cette action permet de lancer un ou plusieurs exécutables sur le terminal de façon **synchrone** : le client Symphony attendra que l'application lancée retourne avant de traiter l'action suivante (une autre section ou une autre application à lancer au sein de l'action en cours).



### SECTION DYNAMIQUE

---

Élément	Définition
Compteur	FileCount
Lot	<ul style="list-style-type: none"><li>FileXX</li></ul>

### 5.6.1. FileCount

Il s'agit du compteur de la section dynamique qui indique le nombre d'application à lancer pour cette action.

### 5.6.2. stoponerror

Idem que pour la section précédente.

### 5.6.3. FileXX

Il s'agit du chemin absolu de l'application à lancer ou d'un fichier associé à une action auprès du Shell de Windows CE/Mobile.

#### Note

*Si un chemin (source ou cible) comporte des espaces (par exemple \Program Files), il n'est pas nécessaire de l'entourer de guillemets dans un fichier CPY.*

*Il est possible d'utiliser les variables d'environnement listées au début du paragraphe [5 Syntaxe des scripts wdp](#).*



## 5.7. ACTION-EXE

Cette action permet de lancer un ou plusieurs exécutables sur le terminal de façon **asynchrone** : le client Symphony n'attendra pas que l'application lancée retourne avant de traiter l'action suivante (une autre section ou une autre application à lancer au sein de l'action en cours) et passera directement à l'exécutable suivant au sein de la section ou à l'action suivante s'il s'agit du dernier élément à lancer.



### SECTION DYNAMIQUE

---

Élément	Définition
Compteur	FileCount
Lot	<ul style="list-style-type: none"><li>File<b>XX</b></li></ul>

### 5.7.1. FileCount

Il s'agit du compteur de la section dynamique qui indique le nombre d'application à lancer pour cette action.

### 5.7.2. stoponerror

Idem que pour la section précédente.

### 5.7.3. File**XX**

Il s'agit du chemin absolu de l'application à lancer ou d'un fichier associé à une action auprès du Shell de Windows CE/Mobile.

## 5.8. ACTION-REGEXP

Cette action permet d'exporter une partie de la base de registre dans un fichier.



### SECTION DYNAMIQUE

---

Élément	Définition
Compteur	FileCount
Éléments fixes	<ul style="list-style-type: none"><li>stoponerror</li></ul>
Lot	<ul style="list-style-type: none"><li>File<b>XX</b></li></ul>

### 5.8.1. FileCount

Il s'agit du compteur de la section dynamique qui indique le nombre d'exportations de la base de registre à lancer pour cette action.

### 5.8.2. stoponerror

Idem que pour la section précédente.

### 5.8.3. FileXX

Il s'agit du fichier à générer suite à l'exportation suivi de la clé ou la valeur à exporter avec la syntaxe suivante:

```
fichier [-a] reg-key
```

où :

- **fichier** : le chemin absolu du fichier à générer
- **[-a]** : option qui permet d'ajouter (append) le résultat de l'exportation à la fin du fichier
- **reg-key** : la clé ou la valeur à exporter. Les chemins qui concernent une clé de la base de registre sont toujours entourés de crochets. Pour exporter une valeur, il suffit d'exclure la valeur des crochets : par exemple `[HKLM\Comm]\Settings` désigne la valeur Settings de la clé `HKEY_LOCAL_MACHINE\Comm`. Pour exporter une clé (et ses sous-clés), il faut donc entourer son chemin de crochets : par exemple `[HKLM\Comm]` désigne la clé `HKEY_LOCAL_MACHINE\Comm` et ses sous-clés. Il est possible d'utiliser les abréviations usuelles suivantes pour les clés racines :

Clé racine	Abbréviation
HKEY_LOCAL_MACHINE	HKLM
HKEY_CURRENT_USER	HKCU
HKEY_CLASSES_ROOT	HKCR
HKEY_USERS	HKU

## 5.9. ACTION-DEL

Cette action permet d'effacer des fichiers. Elle n'effacera pas les répertoires.



### SECTION DYNAMIQUE

Élément	Définition
Compteur	FileCount
Lot	<ul style="list-style-type: none"> <li>▪ FileXX</li> </ul>

### 5.9.1. FileCount

Il s'agit du compteur de la section dynamique qui indique le nombre de fichiers à traiter.

### 5.9.2. stoponerror

Idem que pour la section précédente.

### 5.9.3. FileXX

Chemin absolu du fichier à effacer. Ce chemin peut contenir des caractères jokers (\* et ?) et servir de filtre.

## 5.10. ACTION-REMOVE

Cette action permet d'effacer des fichiers et/ou des répertoires.



### SECTION DYNAMIQUE

Élément	Définition
Compteur	FileCount
Lot	<ul style="list-style-type: none"> <li>▪ FileXX</li> </ul>

### 5.10.1. FileCount

Il s'agit du compteur de la section dynamique qui indique le nombre de fichiers à traiter.

### 5.10.2. stoponerror

Idem que pour la section précédente.

### 5.10.3. FileXX

Chemin absolu du fichier à effacer. Ce chemin peut contenir des caractères jokers (\* et ?) et servir de filtre.\*

## 5.11. ACTION-KILL

Cette action permet de tuer un ou plusieurs processus sur le terminal.



### SECTION DYNAMIQUE

Elément	Définition
Compteur	FileCount
Eléments fixes	<ul style="list-style-type: none"><li>stoponerror</li></ul>
Lot	<ul style="list-style-type: none"><li>FileXX</li></ul>

### 5.11.1. FileCount

Il s'agit du compteur de la section dynamique qui indique le nombre de fichiers à traiter.

### 5.11.2. stoponerror

Idem que pour la section précédente.

### 5.11.3. FileXX

Il s'agit de l'exécutable à stopper. Ce paramètre peut être le chemin complet vers un fichier exécutable ou simplement son nom. Si vous spécifiez un nom simple de fichier (sans le chemin complet), le client cherchera le premier processus qui coïncide avec ce nom parmi les processus en exécution.

Il est recommandé de ne spécifier un nom de fichier simple si il est sûr qu'il y a un seul processus en exécution avec ce nom.

#### Exemple:

```
[action-kill]
filecount = 1
file00 = \flash\myapp.exe
```

## 5.12. ACTION-DELAY

Cette action permet de faire une temporisation (pause) sur le terminal.

### 5.12.1. delay

Durée de la pause en **millisecondes**.

#### Exemple:

```
[action-delay]
delay = 3000
```

### 5.13. ACTION-BOOT

Cette action effectue un warmboot ou un coldboot. Elle peut apparaître une seule fois dans un fichier et elle est toujours la dernière action exécutée.

#### 5.13.1. type

Définit le type de reset à effectuer : cold ou warm.

Valeurs possibles	Définition	Syntaxe
warmboot	Un démarrage à chaud sera effectué (Warm boot)	Type=warmboot
coldboot	Un démarrage à froid sera effectué (Cold boot)	Type=coldboot

#### 5.13.2. immediate

Cette option permet de choisir si le redémarrage demandé sera ou non différé à la fin de tous les scripts wdp à traiter ou s'il sera traité à la fin du script wdp qui le contient.

Valeurs possibles	Définition	Syntaxe
yes	Le reset sera effectué à la fin du script en cours	Immediate=yes
no	Le reset sera effectué après l'exécution de tous les scripts pris en compte	Immediate=no

#### 5.13.3. delay

Permet de mettre une temporisation exprimée en **secondes** avant d'exécuter le reboot demandé.

## 6. Recommandations pour le paramétrage

Ce paragraphe donne des directives générales pour paramétrer efficacement le client Symphony.

### 6.1. Déploiements automatiques

Une des fonctionnalités les plus importantes de Symphony est le déploiement automatique de packages. Cette fonctionnalité se paramètre via la section [AutoUpdate](#) et [Deployment](#). Dans la mesure où seuls les commandes de type UPDATE donnent lieu au déploiement automatique de packages, il faut s'assurer que le terminal envoie ce type de commande de façon automatique. Dans le cas contraire, seule une intervention manuelle assurera le déploiement automatique de packages (par SpnUserBox, onglet Symphony → bouton "Update" ou par la Console Web Symphony).

Voici les moyens possibles pour assurer le déploiement automatique de packages:

- **Au démarrage** : Il est possible de lancer une connexion au serveur Symphony de façon automatique au démarrage de l'application avec les paramètres [DoStartupCommand](#) et [StartupCommand](#). La commande en question devra comporter un UPDATE (U ou LU pour le paramètre [StartupCommand](#)).
- **De façon périodique** : Il est possible de lancer une connexion au serveur Symphony de façon périodique (à intervalles réguliers) avec les paramètres [DoPeriodic](#) et [PeriodicCommand](#). La commande en question devra comporter un UPDATE (U ou LU pour le paramètre [PeriodicCommand](#)).
- **De façon programmée** : Il est possible de lancer une connexion au serveur Symphony de façon programmée (à heure fixe de façon quotidienne) avec les paramètres [DoSchedule](#) et [ScheduleCommand](#). La commande en question devra comporter un UPDATE (U ou LU pour le paramètre [ScheduleCommand](#)).
- **Lors de l'établissement d'une connexion réseau** : Il est possible de lancer une connexion au serveur Symphony à la détection de l'établissement d'une connexion réseau avec les paramètres [OnConnect](#) et [OnConnectCommand](#). La commande en question devra comporter un UPDATE (U ou LU pour le paramètre [OnConnectCommand](#)).  
 Cette dernière possibilité est à réserver pour une connexion non permanente (type 3G/GPRS). En effet, une connexion permanente (type Wifi/ethernet) est établie au démarrage du terminal une fois pour toutes et les événements de connexions détectés font suite à des problèmes de déconnexions intempestives non souhaitées (terminal hors zone de couverture). Dans ce cas, les événements de connexions réseau auront une nature peu prévisibles et ne constituent pas une base fiable pour le déploiement automatique. En revanche, cette fonctionnalité de commande automatique convient assez bien à une connexion de type 3G/GPRS qui est établie/fermée à la demande.

### 6.2. Optimisation des commandes périodiques

Toute commande (LIST ou UPDATE) issue d'un terminal donne lieu à une intégration côté console Symphony: les données envoyées sont lues et insérées dans la base de données sous-jacente (ce qui permet d'alimenter l'historique et le journal de chaque terminal sur la console). Afin de ne pas surcharger la Console et de garder des performances acceptables, il convient d'adapter la fréquence des commandes périodiques et de ne pas paramétrer une période trop basse. Le tableau ci-dessous indique les valeurs de période minimales pour les commandes périodiques en fonction du nombre de terminaux connectés à une même console:

Nombre de terminaux	Période préconisée (en min)
Entre 1 et 200	5
Entre 200 et 1000	30
Plus de 1000	60

### 6.3. Répertoire d'installation

Le client Symphony doit être installé dans un répertoire nommé `wdp`.