ExecAction

Copyright © 2015, ND-Faciliti

Table des matières

ND-Faciliti ExecAction 3
Balises 4
Standard 5
CHECK 5
LOCALUSERGROUP 13
SCREENSHOT 15
MEMO 17
CREATE 21
COPY 23
DELETE 25
EXEC 27
MAIL 29
MSGBOX 32
RENAMEDISK 35
REGISTRY 36
SHORTCUT 39
SESSION 41
PAUSE 43
INPUTBOX 44
FILEBOX 46
PATHBOX 47
HTTP 49
WAIT 51
LOG 55
WHILE 57
FOR 59
LIB 61
PLUGIN 64
SYSTRAY 67
EXPORT 70
Spécifique 72
Racine action 72
ACTION 75
COMOGUI 79
WINDOW 80
EXECACTION 84
Fichier de directive XML 86
Légende des icônes 87
Utilisation de l'executable 88
Exemple de scripts 89
Gestion des erreurs 91
Déclenchement des actions 94
Mémo constantes 96
Construction de module complémentaires 97
Historique des versions 98
Légende de notation 104

ND-Faciliti ExecAction

ND-Faciliti ExecAction

ExecAction est un séquenceur d'opérations conditionnées. Il à été conçu pour répondre au besoin simple d'un script de login, installation d'application, mise en conformité de machines,GPO Windows, que ce soit dans l'environnement *utilisateur* ou *machine*. Il permet entre autre d'automatiser des tâches de gestion de parc de la "masterisation" à la maintenance du parc informatique, tel que le séquençage et l'installation d'applications.

Les fichiers de directives sont de format XML.

Une rubrique donne des liens pour l'explication du format XML.

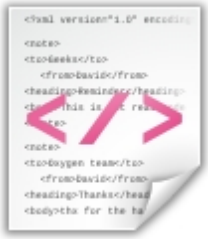
Version : 4
Révision : 10
Mode : Stable

Plusieurs versions sont disponible (x64/x86).

Créé avec HelpNDoc Personal Edition: Générateur de documentation complet

Balises

Description des balises



Cette partie décrit chaque balises et comment les utiliser.
Elle est divisé en plusieurs catégories pour mieux les repérer.

En général il existe un exemple par balise.

Dans certains attributs, il est possible de spécifier des variables d'environnements.
la chaîne pour les interpréter sera par exemple, #computername#.

Notation dans les tableaux des attributs

Pour chaque balise, il y des attributs qui sont consignés dans un tableau récapitulatif divisé en 3 colonnes.

Note sur la notation des directives :

: Paramètre obligatoire.

: Paramètre optionnel.

: En fonction de... (d'un type, d'un mode).

: Possibilité d'exécuter des chaînes d'abstraction.

: Valeur fixe ou syntaxe fixe.

: Information sans obligation.

: Information importante à prendre en compte.

Créé avec HelpNDoc Personal Edition: Générateur complet de livres électroniques Kindle

Standard

Balises standard

Ici sont décrites les balises qui sont utilisé pour exécuter des actions.

Ce sont des directives.

Pour chaque directive une description et une explication des attributs est effectuée.

Certaines directives (balises), utilisent des variables d'environnement, sur celles-ci certaines variables peuvent être génératrice d'effet de bord indésirable.

Exemple : %temp%\toto.log sera traduit par C:\DOCU~1\...

Dans la directive MEMO, il sera impossible d'ouvrir ce fichier, car il n'existe pas à cette emplacement. En effet une étape de vérification ne pourras pas ouvrir le document car son vrai emplacement se trouve c:\document and settings\....

Créé avec HelpNDoc Personal Edition: Créer des documents d'aide PDF facilement

CHECK

CHECK

Description :

La balise CHECK sert pour la vérification aussi bien en externe (fichiers, répertoires, processus) ou en interne (code retour de la balise EXEC ou ACTION).

Elle sert donc à conditionner certains passages

Contrairement à un IF en programmation, elle ne comporte pas de ELSE.

L'attribut "name" est sensible à la casse.

Type des attributs :

Obligatoire :

type : Détermine le type de structure a vérifier.

value : Valeur à vérifier. Que ce soit pour un fichier en mode MD5 ou taille ou date de modification, nom d'un groupe, clef de base de registre ou version de système d'exploitation.

mode: Pour certaines vérification cette option est obligatoire.

Optionnel (en fonction du contexte) :

name : Permet de spécifier le nom d'une étiquette afin d'en vérifier la valeur.

mode : Configure le mode de vérification du type spécifié (type). Peut être obligatoire pour certaines vérifications.

location : Emplacement à vérifier ou chemin fichier ou emplacement et / ou clef dans le registre. Paramètre pouvant accepter les valeurs de type "expand".

blockTag : Condition de passage dans la balise check.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**file** : vérifie un fichier. **directory** : vérifier un répertoire. **ADgroup** : vérifier un groupe Active Directory (mode utilisateur). **registry** : vérifie dans la base de registre. **sysver** : version du système d'exploitation. **arch** : Type d'architecture processeur. **process** : vérifie un nom de processus en mémoire. **date** : Vérifie par rapport à une date ou une période. **exec** : Vérification du code retour d'une exécution d'un programme. **action** : Vérification l'état de l'action. **memo** : Vérification d'un contenu de mémorisation. **freespace** : Espace disque. **msgbox** : Retour des boites à message où l'utilisateur a cliquer sur un bouton. **lib** : Retour des boites à message où l'utilisateur a cliquer sur un bouton.	Pour le type exec, action, memo : Les options *true, false et ignored*
Name	Pour le type ***exec*** : Nom de la directive exec qui a	Inscrire le nom de l'étiquette du nom de l'action ou de l'exec.

	été lancé. Pour le type ***action*** : Nom de l'action à vérifier	
Mode	Pour le type ***file*** : **test** : mode par défaut, teste juste la présence. **md5** : Vérification en MD5 **size** : Vérification en taille du fichier (octets) Pour le type ***date*** : ***punctual*** : mode par défaut, test une date ponctuelle ou une série de date ***period*** : test la date sur une période de temps. ***stopdate*** : Si la date dépasse celle indiqué ***startdate*** : à partir de la date. Pour le type ***exec, action, memo, lib, arch*** : ***equal*** : mode par défaut, test le code retour si égale. ***dif*** : test si la valeur est différente de la valeur à tester. ***cont*** : test si la valeur contient la valeur à tester. Ce mode fait une recherche de chaîne de caractères Pour le type ***msgbox*** : ***equal*** : mode par défaut, test le code retour si égale. ***dif*** : test si la valeur est différente de la valeur à tester. Pour le type ***registry*** : ***key*** : test une clef de base de registre. ***value*** : test une valeur de base de registre.	
Location		Les valeurs peuvent contenir

	Pour le type ***registry*** : Emplacement de la clef ou de la valeur. Pour le type ***file*** : Chemin du fichier. Pour le type ***freespace*** : Unité à verifier.	des chaînes d'abstraction.
Value	Pour le type ***sysver*** : **win_nt351** **win_nt4** **win_95** **win_98** **win_2000** **win_xp** **win_7** **win_7_32** **win_7_64** **win_vista** **win_vista_32** **win_vista_64** **win_8** **win_8_32** **win_8_64** **win_8.1** **win_8.1_32** **win_8.1_64** **win_s2003** **win_s2008** **win_s2012** **win_me** **win_other** **win_10** **win_10_32** **win_10_64** Pour le type ***process*** : Le nom d'exécutable. Pour le type ***registry*** : Contenu de la valeur si besoin pour le test. Pour le type ***ADgroup*** : Le nom du groupe ou des	Pour le type ***registry*** : le contenu de la valeur n'est pas obligatoire. Si rien n'est spécifié, ça sera juste la valeur ou la clef qui sera testé. Pour le type ***ADgroup*** : Attention pour spécifier un nom de groupe, il faut obligatoirement charger les groupes sur la "racine action". Il est possible de spécifier plusieurs nom de groupe séparé par des virgules. Si l'utilisateur fait parti d'un groupe alors le vérification est validé. Pour le type ***date*** : En mode **period, StopDate, StartDate** : Il est possible de spécifier

	groupes Pour le type ***file*** : la valeur du MD5 ou de la taille en octets. Pour le type ***date*** : En mode **punctual** : "**jj/mm/aaaa**". Il est possible de placer plusieurs dates séparées par une virgule. En mode **period** : "**jj/mm/aaaa**". Il est possible de placer périodes de dates séparé par une virgule. En mode **stopDate** : "**jj/mm/aaaa**". Il n'est pas possible de placer plusieurs date. En mode **startDate** : "**jj/mm/aaaa**". Il n'est pas possible de placer plusieurs date. Pour le type ***exec*** : **valeur à comparer (code retour du programme)** Pour le type ***lib*** : **valeur à comparer (code retour de la fonction)** Pour le type ***action*** : **valeur à comparer (true, false, ignored)** Pour le type ***msgbox*** : **valeur à comparer (yes, no, ok, cancel, continue, retry)** Pour le type ***arch*** : **all** **x86** **x64** **amd64**	plusieurs date, il faut séparer par des virgules les valeurs. (ex: *jj/mm/aaaa,jj/mm/aaaa*). Pour le type ***action*** : Les options *true, false et ignored* correspondent à *1, 0 et -2.* Ne pas utiliser à l'intérieur de l'action que l'on vérifie. En effet celle-ci n'étant pas terminé, il ne sera pas possible de pouvoir la vérifier au niveau du retour. Pour le type ***exec*** : Il est possible de spécifier plusieurs code retour d'exécutions. Pour le type ***lib,arch,sysver*** : Il est possible de spécifier plusieurs codes retour d'exécutions avec des "," Pour le type ***arch,sysver*** : Il est possible de spécifier plusieurs valeurs à tester ","

BlockTag	**0** ou **pie** : Passage si existe (par défaut). *pie : Pass If Exist* **1** ou **pne** : Passage si n'existe pas. *pne : Pass if Not Exist* **2** ou **signalerror** : Remonte en erreur si l'expression (à l'aide de mode, type et value) n'est pas "vrai".	Si l'attribut n'est pas spécifié l'option sera automatiquement générée à "pie" par défaut.

Exemple :

Pour vérifier un fichier en mode MD5 :

```
<CHECK type="file" mode="md5" location="#temp#\essai.xml"
value="dsdf6s51sdfsd5651sfd">
    ...
    ...
</CHECK>
```

Pour vérifier un répertoire :

```
<CHECK type="directory" location="c:\" blockTag="1">
    ...
    ...
</CHECK>
```

Pour vérifier une appartenance dans un groupe de l'utilisateur :

```
<CHECK type="ADGroup" value="SUPER_GROUPE,MON_DEUXIEME_GROUPE">
    ...
    ...
</CHECK>
```

Pour vérifier la version du système d'exploitation :

```
<CHECK type="sysver" value="win_2000,win_xp">
```

```
    ...
    ...
</CHECK>
```

Pour vérifier le code retour d'une application lancé avec la balise EXEC :

```
<CHECK type="exec" name="MonProgrammeLancement" value="0,3010">
       ...
       ...
</CHECK>
```

Pour vérifier la date ou une période de temps :

```
<CHECK type="date" mode="startdate" value="19/10/2010">
       <MSGBOX title="startDate" text="Date valide" type="ok" />
</CHECK>

<CHECK type="date" mode="stopdate" value="19/10/2010">
       <MSGBOX title="stopDate" text="Date valide" type="ok" />
</CHECK>

<CHECK type="date" mode="period" value="17/10/2010|18/10/2010,20/10/2010|
21/10/2010">
       <MSGBOX title="period" text="Date valide" type="ok" />
</CHECK>

<CHECK type="date" value="17/10/2010,18/10/2010,19/10/2010">
       <MSGBOX title="ponctuel" text="Date valide" type="ok" />
</CHECK>
```

Pour conditionner le lancement d'une action :

```
<CHECK type="file" mode="md5" location="#temp#\essai.xml"
value="dsdf6s51sdfsd5651sfd">
       <ACTION name="COPY" version="0" descriptionAction="Essai en expand encore
!">
              <COPY type="file" sourceLocation="F:\Projets\ndf_execAction
       \essai.log"  targetLocation="d:\temp\azertyu\iop\qsd\fghj\toto.txt"
       mode="createDir"/>
       </ACTION>
</CHECK>
```

Créé avec HelpNDoc Personal Edition: Création d'aide CHM, PDF, DOC et HTML d'une même source

LOCALUSERGROUP

LOCALUSERGROUP

Description :

Ajoute, supprime, un utilisateur ou un groupe en local sur la machine.

Cette balise ne concerne pas l'Active Directory, juste les comptes et groupes locaux.

Type des attributs :

Obligatoire :

- **type** : Sélection de cible (groupe, membre d'un groupe, utilisateurs).
- **mode** : détermine les actions a mener (ajout, suppression,check).

Optionnel (en fonction du contexte) :

- **Username** : Nom de l'utilisateur.
- **Password** : Mode de passe de l'utilisateur.
- **GroupName** : Nom du groupe.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?

Type	**group** : Cible le groupe. **member** : Cible le contenu du groupe. **user** : Cible un compte utilisateur.	
Mode	**add** : Ajout un élément. **delete** : Supprime un élément **check** : Verifié un élément	
Username	Pour le type ***user***, ***member*** : ***nom d'utilisateur***	Inscrire le nom de l'utilisateur.
Password	Pour le type ***user*** ***mot de passe***	Inscrire le mot de passe (pouvant être chiffré si besoin).
GroupName	Pour le type ***group*** : ***nom du groupe***	Inscrire le nom du groupe
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

Pour ajouter un utilisateur avec un mot de passe :

```
<LOCALUSERGROUP type="utilisateur" mode="ajout" nomUtilisateur="supercool"
mdpUtilisateur="zezette"/>
```

Créé avec HelpNDoc Personal Edition: Écrire des livres électroniques ePub pour l'iPad

SCREENSHOT

SCREENSHOT

Description :

Permet la capture de l'écran sur lequel se déroule l'exécution du fichier de directive.

La capture peut être en plein écran ou défini par la taille et la hauteur.

Type des attributs :

Obligatoire :

- **FilePath** : Chemin du fichier image.

Optionnel (en fonction du contexte) :

- **Type** : Format du fichier qui sera enregistrée.
- **Mode** : Configuration de la prise de vue.
- **HeightPosition** : Position de départ en hauteur du cadre de la capture.
- **WidthPosition** : Position de départ en largeur du cadre de la capture.
- **Height** : Hauteur de l'image capturée en fonction des positions de départ.
- **Width** : Largeur de l'image capturée en fonction des positions de départ.

- **Ratio** : Taux de compression de l'image.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
FilePath	**Chemin du fichier à enregistrer**	La valeur peut contenir des abstractions de chaînes (ex : C:\%windir% ou C:\#windir#\)
Type	**JPG** : Format JPEG **BMP** : Format BITMAP **PNG** : Format	
Mode	**full** : La capture sera fait en plein écran. **retail** : la capture sera retaillée. (en fonction de la dimension spécifié)	
HeightPosition	format en pixel	Position de départ en hauteur du cadre de la capture.
WitdhPosition	format en pixel	Position de départ en largeur du cadre de la capture.
Height	format en pixel	Dimension en hauteur de l'image capturé.
Witdh	format en pixel	Dimension en largeur de l'image capturé.
Ratio	Pour le type ***JPG*** : **0** à **10** : Le ratio de compression de l'image.	Valeur allant de 0, pour la qualité la plus basse, vers 10, pour la qualité la plus haute.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

	-1 ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	

Exemple :

```
<SCREENSHOT FilePath="c:\toto.jpg" type="png" mode="full" blockTagError="block"/>
```

```
<SCREENSHOT FilePath="c:\toto.jpg" type="bmp" mode="full" blockTagError="ignore"/
>
```

```
<SCREENSHOT FilePath="c:\toto_2.jpg" type="jpg" mode="retail" HeightPosition="20"
WitdhPosition="20" Height="200" Width="200" ratio="10" blockTagError="inherit"/>
```

```
<SCREENSHOT FilePath="c:\toto.jpg" type="jpg" mode="full" ratio="10"
blockTagError="1"/>
```

Créé avec HelpNDoc Personal Edition: Générer des livres électroniques EPub facilement

MEMO

MEMO

Description :

Permet de "mémoriser" de valeurs.

Grâce à ce système il est possible de pouvoir réutiliser des valeurs dans les autre balise, récupéré depuis la base de registre, d'un fichier, d'une valeur interne défini dans le fichier.

Cette balise va chercher l'information et la mémoriser.

Type des attributs :

Obligatoire :

Type : Endroit où se trouve la valeur à récupérer. (base de registre, fichier, interne, etc...)

Name : Nom de l'étiquette servant à mémoriser la valeur récupéré.

Optionnel (en fonction du contexte) :

Value : Valeur à indiquer obligatoirement avec le type *internal*, mais aussi en option avec les autres type.

Location : Emplacement soit de la valeur de base de registre, soir du fichier à ouvrir.

Group : Emplacement soit de la valeur de base de registre, soir du fichier à ouvrir.

BlockTagError : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**internal** : Permet de spécifier une valeur qui pourra être utilisé par la suite. **file** : Récupère le contenu depuis un fichier. **registry** : Depuis la base de registre. **ini** : Charge depuis un fichier INI en mémo. **clipboard** : Récupère le presse papier dans sa totalité en mode texte. **parameters** : Permet de récupérer un paramètre spécifié.	Pour le type ***registry***, il est possible de récupérer des informations de type reg_sz, dword et multisz. Pour les valeurs en multiSZ, il est possible de spécifier le numéro de ligne.

Name	Pour le type ***internal*** : **Nom de l'étiquette** Pour le type ***file*** : **Nom de l'étiquette** Pour le type ***registry*** : **Nom de la valeur** **Nom du préfixe** Pour le type ***ini (optionnel)*** : **Nom de l'étiquette**	De préférence chiffre et lettre seulement. Pour le type ***registry,*** en fonction du mode choisis, il est possible de spécifier soit un nom de valeur, soit de préfixé les mémo. voir l'option Mode. Pour le type ***ini,*** si une étiquette est spécifié, alors il recherchera automatiquement le même nom dans le fichier.
Value	Pour le type ***internal*** : **Valeur à mémoriser** Pour le type ***ini, file*** : **Valeur par défaut si la valeur n'existe pas.** Pour le type ***registry*** : **Valeur si le fichier n'existe pas**	Pour le type ***ini, file*** et ***registry*** il est possible de spécifier une valeur par défaut en cas d'échec de lecture ou d'ouverture des différentes ressources.
Mode	Pour le type ***internal*** : Aucune option pour l'instant. Pour le type ***file*** : Aucune option pour l'instant. Pour le type ***ini*** : **AssoGroupName** : Préfixe les mémos du groupe où elles ont été récupérées. Pour le type ***registry*** : Key : Récupération du plusieurs valeurs dans une clef	Pour le type ***ini***, le préfixe sera du type [NomGROUPE]_[NomVALEUR] Pour le type ***registry***, il est possible de spécifier un nom afin de préfixer les noms de mémo. Voir l'option Name à ce sujet.
Location	Pour le type ***file*** : **Chemin vers le fichier** Pour le type ***ini*** : **Chemin vers le fichier** Pour le type ***registry*** : **Chemin dans la base de registre**	Pour le type ***registry***, Il est possible de récupérer la valeur de la clef par défaut. Le backslash est important après le nom de la clef. Nom / Type / Données (par défaut) / REG_SZ / abcdefgh Exemple : *location="HKLM\SOFTWARE\MYKEY\"*
Line	Pour le type ***file*** : **N° de la ligne du fichier** Pour le type ***registry*** :	

	N° de la ligne du MultiSZ Pour le type ***ini*** : **Nom de la valeur**	
GroupName	Pour le type ***ini*** : **Nom du groupe contenant les valeur et leur contenu**	
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<MEMO type="internal" name="prout" value="Coucou"/>

<MEMO type="registry" location="HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation
\Global\PersistenceIdentifier" name="essai_bdr"/>

<MEMO type="registry" location="HKEY_CURRENT_USER\Software\CodeBlocks\Path"
name="essai_bdr"/>

<MEMO type="file" location="#windir#\windowsupdate.log" name="essai_fichier"/>

<MEMO type="file" location="#windir#\memory.dmp" name="essai_fichier"/>

<MEMO type="file" location="#windir#\windowsupdate.log" name="ligne_fichier"
line="15"/>
```

Créé avec HelpNDoc Personal Edition: Éditeur complet de livres électroniques ePub

CREATE

CREATE

Description :

Balise permettant la création d'un fichier avec du texte ou non ou un répertoire.

Type des attributs :

Obligatoire :

- **Type** : Type de structure à créer.

Optionnel (en fonction du contexte) :

- **Mode** : Configuration de la prise de vue.
- **GroupName** : Nom du groupe.
- **Location** : Chemin du fichier ou du répertoire.
- **Value** : Chemin du fichier ou du répertoire.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**directory** : Créer un répertoire.	La valeur peut contenir des abstractions de chaînes (ex : C:\%windir% ou C:\#windir#\)

	file : Créer un fichier. **Evariable** : Créer une variable d'environnement.	
Mode	Pour le type ***file*** : **replace** : Ecrase le fichier. **ascii** : utilise le BOM ascii. **utf8** : utilise le BOM utf8. **unicode** : utilise le BOM utf8. **rc** : Utilise le retour à ligne.	Les valeurs peuvent être cumuler en les séparant avec des virgules.
Name	Pour le type ***evariable*** : **Nom de la variable**	
Location	**Chemin du fichier ou répertoire**	
Value	Pour le type ***file*** ou ***evariable*** : **Valeur à intégrer**	: MEMO, EXEC, retour action.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<CREATE type="directory"  location="d:\temp\ouf"/>
```

```
<CREATE type="file"  mode="replace" location="d:\temp\supercreer.txt"
value="#username# coucou c'est super #date#"/>
```

```
<CREATE type="file"  mode="replace,utf8" emplacement="d:\temp\supercreer.txt"
value="#username# coucou c'est super #date#   -- 2"/>
```

Créé avec HelpNDoc Personal Edition: Créer des documentations web iPhone

COPY

COPY

Description :

Opération de copie en masse ou unitaire, de répertoires ou de fichiers.

Type des attributs :

Obligatoire :

- **Type** : Détermine le type de copie (répertoire ou fichier).
- **SourceLocation** : Emplacement ou fichier source.
- **Targetocation** : Emplacement ou fichier cible.

Optionnel (en fonction du contexte) :

- **Mode** : Différents permettant de remplacer, écraser, etc.
- **FileFilter** : Filtre sur fichier à copier
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**file** : Copie de fichier **directory** : Copie de répertoire	
SourceLocation	**Emplacement ou fichier source.**	
TargetLocation	**Emplacement ou fichier cible.**	
Mode	Pour le type ***directory*** : **recursive** : Copie les sous-répertoires. **replace** : Remplace si existant (mode par défaut) Pour le type ***file*** : **createdir** : Tente de créer l'arborescence si celle-ci n'est pas crée.	Les options peuvent être cumulé par une virgule.
FileFilter	Pour le type ***directory*** : **xxxxx.xxx** : Filtre le nom ou les types de fichiers.	Par défaut tout les fichiers seront pris.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<COPY type="file" sourceLocation="E:\Projets\ndfaaaa_execAction\essai.xml"
targetLocation="#temp#\essai.xml" />
```

Créé avec HelpNDoc Personal Edition: Générateur complet de livres électroniques Kindle

DELETE

DELETE

Description :

Supprime répertoires ou fichiers.

Type des attributs :

Obligatoire :

- **Type** : Détermine le type de structure à supprimé.
- **Location** : Emplacement source pour le fichier ou répertoire.

Optionnel (en fonction du contexte) :

- **Mode** : Différents permettant de remplacer, écraser, etc.
- **FileFilter** : Filtre sur fichier à copier
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?

Type	**file** : suppression de fichier. **directory** : suppression de répertoire. **cleandir** : suppression du contenu d'un répertoire.	
Location	**Emplacement du répertoire ou du fichier.**	Par défaut tout les fichiers seront pris.
Mode	Pour le type ***directory*** : **recursive** : parcours les sous répertoires. *(pour les sous répertoires)* **replace** : Remplace si existant (mode par défaut)	Les options peuvent être cumulé par une virgule. Pour le type ***cleandir*** : Les fichiers et sous répertoires seront nettoyés.
FileFilter	Pour le type ***file*** : **xxxxx.xxx** : Filtre le nom ou les types de fichiers.	Par défaut tout les fichiers seront pris.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<DELETE type="file" location="E:\Projets\ndfaaaa_execAction\essai.log" blockTagError="inherit"/>
```

```
<DELETE type="file"  location="E:\Projets\ndfaaaa_execAction.log" blockTagError="ignore"/>
```

```
<DELETE type="directory" mode="recursive" fileFilter="*.log" location="E:\Projets\ndfaaaa_execAction"/>
```

Créé avec HelpNDoc Personal Edition: Créer des livres électroniques EPub facilement

EXEC

EXEC

Description :

Lancement d'un programme externe.

Type des attributs :

Obligatoire :

Type : Méthode de lancement.

ProgramLocation : Emplacement source pour le fichier ou répertoire.

Optionnel (en fonction du contexte) :

Name : Nommage de la directive afin de mémoriser le code retour pour un traitement ultérieur.

Mode : Règle l'exécution du programme (caché,attente).

- **WaitingTime** : Temps d'attente maximum du programme.
- **Parameters** : Paramètres de l'exécutable à lancer.
- **WorkingDirectory** : Répertoire de travail ou répertoire courant.
- **ValidReturnCode** : Indique les codes retours qui servirons à valider le bon retour ou non de l'application lancée.
- **Output** : Chemin d'un fichier se sortie du texte retour de l'application lancée.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**console** : Pour les programmes en mode console. **window** : programme fenêtré.	
ProgramLocation	**Emplacement source pour le programme.**	
Name	**Nom alphanumérique.**	Grâce au nommage de l'action exec, il est possible de mémoriser le code retour de l'application pour un traitement ultérieur avec la balise CHECK.
Mode	**hidden** : Mode caché l'application ne sera pas visible. **wait** : L'application est attendu.	Les options peuvent être cumulé séparé par une virgule.
WorkingDirectory	**Répertoire de travail ou répertoire courant.**	
Parameters	**Paramètres optionnel pour l'exécutable à lancer.**	
ValidReturnCode	**Code retours de l'application**	Les valeurs peuvent être cumulé

	qui seront valide	séparé par une virgule.
WaitingTime	Pour le mode ***wait***: **Temps d'attente en secondes.**	Si l'application retourne du texte, il sera possible de le stocker dans un fichier.
Output	**Chemin du fichier de sortie texte.**	Si l'application retourne du texte, il sera possible de le stocker dans un fichier.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<EXEC name="notepad" programLocation="#windir#\notepad.exe" mode="hidden,wait"
waitingTime="10" validReturnCode="0" blockTagError="ignore"/>
```

Créé avec HelpNDoc Personal Edition: Outils facile d'utilisation pour créer des aides HTML et des sites web

MAIL

MAIL

Description :

Envois un message mail via SMTP avec possibilité d'attacher des fichiers. Il n'y a pas d'authentification possible pour l'instant.

Type des attributs :

Obligatoire :

- **Server** : Nom ou adresse IP du serveur de messagerie SMTP
- **To** : Adresses de messagerie des destinataires principaux.

Optionnel (en fonction du contexte) :

- **Type** : Format du message.
- **SenderName** : Nom de l'expéditeur du message. Par défaut il sera vide.
- **SmtpPort** : Port du serveur SMTP.
- **Cc** : Adresse de messagerie des destinataires en copie.
- **Bcc** : Adresse de messagerie des destinataires en copie caché.
- **Object** : Sujet du message.
- **Contenu** : Corps du message.
- **Attachement** : Chemin de fichiers attachés au message.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Server	**Adresse IP ou nom**	S'assurer de l'autorisation sur le serveur pour l'envoi de message
To	**Adresse de messagerie des destinataires principaux.**	Les adresses de messagerie des destinataires principaux, doivent être séparées par une virgule. : MEMO

Type	**ascii** : Le message sera en mode ascii (par défaut). **utf-8** : Le message sera en mode UTF-8.	
Cc	**Adresse de messagerie des destinataires en copie.**	Les adresse de messagerie des destinataires principaux, doivent être séparées par une virgule. : MEMO
Bcc	**Adresse de messagerie des destinataires en copie caché.**	Les adresse de messagerie des destinataires principaux, doivent être séparées par une virgule. : MEMO
Object	**Sujet du message**	: MEMO, variable externes.
Content	**Corps du message**	: MEMO, variable externes, code retour d'actions.
Attachement	**Emplacement fichier**	Attache des fichiers. Il est possible de remplacer le nom du fichier attaché par un nom de description (il faut séparer le chemin du fichier par **|**. Par défaut le nom de fichier sera repris. Il est possible d'ajouter plusieurs fichier en séparant par des virgules. : MEMO, variable externes.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

	d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	

Exemple :

```
<MAIL server="10.1.1.1"
          senderName="#computername#"
to="monadresse@ndfaciliti.com,toto@ndfaciliti.com"
object="coucou" content="Bonjour et bienvenu !"
attachment="c:\temp\config.xml|Mon_super_fichier,d:\temp
\ndfLogin.jpg|Image_genial"/>
```

Créé avec HelpNDoc Personal Edition: Créer des aides HTML, DOC, PDF et des manuels depuis une même source

MSGBOX

MSGBOX

Description :

Affiche un fenêtre de type afin d'avertir l'utilisateur d'une action.

Type des attributs :

Optionnel (en fonction du contexte) :

- **Title** : Titre de la fenêtre
- **Text** : Texte qui sera le contenu du message.
- **Type** : Type de bouton sur la fenêtre.
- **IconType** : Type d'icône.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Title	**Titre de la fenêtre**	
Text	**Texte de la fenêtre**	: MEMO, variable externes, code retour d'actions.
Name	**Nom du mémo**	il sera donc possible de vérifier le retour de la fenêtre avec la balise CHECK
Type	**ok** : Affiche le bouton "OK" **yesno** : Affiche les boutons "Yes/No" ou "Oui/Non". **yesnocancel** : Affiche les boutons "Yes/No/Cancel" ou "Oui/Non/Annuler".	Option non cumulable. En spécifiant cette option, il est possible ensuite de conditionner le retour de la directive.
IconType	**stop** : Affiche un icône **question** : Affiche un icône **warn** : Affiche un icône **notify** : Affiche un icône	Option non cumulable.

BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<MSGBOX title="Mon super titre" text="Coucou !" type="ok"/>

<MSGBOX name="MaFen" title="Mon super titre" text="Coucou !" type="yesnocancel"/>
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<DEMO>
	<MSGBOX title="This is a demo" text="This is a demo for message box"
type="ok"/>

	<MSGBOX name="Mychoice1" title="This is a demo [2]" text="What do you want
?" type="yesno" IconType="question"/>

	<MSGBOX name="Mychoice2" title="This is a demo [3]" text="Cancel or not ?"
type="yesnocancel" IconType="question"/>

	<CHECK Type="msgbox" name="Mychoice1" value="yes">
		<MSGBOX title="Result for Choice1" text="result yes (#Mychoice1#)"
type="ok" IconType="notify"/>
	</CHECK>

	<CHECK Type="msgbox" name="Mychoice1" value="yes" mode="dif">
		<MSGBOX title="Result for Mychoice1" text="result is not yes
(#Mychoice1#)" type="ok" IconType="notify"/>
	</CHECK>

	<CHECK Type="msgbox" name="Mychoice2" value="cancel" mode="dif">
		<MSGBOX title="Result for Choice1" text="result is not cancel
(#Mychoice2#)" type="ok" IconType="notify"/>
```

```
        </CHECK>

</DEMO>
```

Créé avec HelpNDoc Personal Edition: Générateur complet de livres électroniques Kindle

RENAMEDISK

RENAMEDISK

Description :

Renomme le nom d'un volume.

Type des attributs :

Obligatoire :

- **Drive** : Lettre de lecteur.
- **NewName** : Nom pour le changement de nommage.

Optionnel (en fonction du contexte) :

- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Drive	**Lettre de lecteur suivis de :**	La lettre de lecteur doit être suivis systématiquement de **:/**

NewName	**Nom alphanumérique**	: MEMO, variable externes, code retour d'actions.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<RENAMEDISK drive="d:\" newName="Données"/>
```

Créé avec HelpNDoc Personal Edition: Générateur complet d'aides multi-formats

REGISTRY

REGISTRY

Description :

Manipulation dans la base de registre.

Type des attributs :

Obligatoire :

- **Location** : Emplacement de la valeur ou clef.

Optionnel (en fonction du contexte) :

- **Type** : Type de valeur.
- **Mode** : Opération d'ajout ou de suppression.
- **ValueName** : Nom de la valeur ou valeur par défaut de la clef.
- **EnableRegistry64** : Enlever la redirection registre 64bits.
- **Value** : Valeur.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Location	**Emplacement de la valeur**	Emplacement sans le nom de valeur. Lors d'une création de clef, il faut spécifier aussi le nom de clef dans l'emplacement.
Type	**reg_dword** **reg_sz** **reg_multisz** : Ajoute à la suite des valeurs déjà présentes. **reg_binary** **reg_expand_sz**	par défaut *reg_none*.
Mode	**add** : Ajoute la valeur ou la clef **del** : Supprime la valeur **delkey** : Supprime la clef	La suppression en mode *delkey* entraîne automatiquement la suppression de toutes valeurs et sous-clefs présentent dans le parent.
EnableRegistry64	**False** (héritage par défaut) :	

	Désactive la redirection registre 64Bits. **True** : Active la redirection 64Btis	
ValueName	**Nom de la valeur**	Si le nom de la valeur est laissé vide, alors le contenu de la valeur, sera automatiquement attribué à la valeur par défaut de la clef spécifié par l'option "*location*".
Value	**Contenu de la valeur**	: MEMO, variable externes, code retour d'actions.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<REGISTRY mode="add" type="reg_sz" location="HKEY_CURRENT_USER\Software\ND-
Faciliti" valuename="essai_dans_regsitre" value="bonjour !"
blockTagError="block"/>
```

```
<REGISTRY mode="delkey" location="HKEY_LOCAL_MACHINE\SOFTWARE\SOURCE"/>
```

Créé avec HelpNDoc Personal Edition: Créer des documents d'aide CHM facilement

SHORTCUT

SHORTCUT

Description :

Opération de création de raccourcis.

Type des attributs :

Obligatoire :

Location : Emplacement de création du raccourcis.

Optionnel (en fonction du contexte) :

TargetPath : Emplacement où point le raccourci.

Parameters : paramètres pour la cible.

Comment : Commentaire du raccourcis (tooltip).

WorkingDirectory : Emplacement du répertoire de travail.

OpenMode : Sert à régler paramètre de méthode de lancement de l'application.

FileIcon : Spécifie un autre fichier contenant les icônes.

FileIconIndex : Numéro de l'icône indexé dans le fichier.

BlockTagError : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Location	**Chemin où sera créer le raccourcis.**	Emplacement avec le nom du fichier
TargetPath		: MEMO, variable externes

	Chemin cible du programme à lancer	
Parameters	**Paramètre pour le programme à lancer**	Les options peuvent être séparé par une virgule.
Comment	**Commentaire du raccourcis alphanumérique**	
WorkingDirectory	**Valeur de KEY**	: MEMO, variable externes, code retour d'actions.
OpenMode	**maxi** : Lancement maximisé **mini** : Lancement minimisé **normal** : normal (par défaut)	
FileIcon	**Chemin du fichier contenu l'icône.**	
FileIconIndex	**De 1 à 99999** : Index (position) dans le fichier icône.	
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<SHORTCUT location="c:\essai.lnk" targetPath="#windir#\notepad.exe" comment="Essai de creation de raccourcis" workingDirectory="%windir%" openMode="maxi" fileIcon="#windir#\twunk_16.exe" fileIconIndex="2"/>
```

Créé avec HelpNDoc Personal Edition: Créer des aides HTML, DOC, PDF et des manuels depuis une même source

SESSION

SESSION

Description :

Opération qui agit l'environnement de l'exécution du fichier de directive.

Type des attributs :

Obligatoire :

- **Type** : Type de service demandé.

Optionnel (en fonction du contexte) :

- **Value** : Valeur de retour.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**reboot** : Demande de redémarrage de la machine **shutdown** : Demande d'arrêt de la machine **lock** : Vérouillage session utilisateur **logoff** : Déconnexion session utilisateur **quit** : Arrête ExecAction **actionquit** : Sort de force de l'action courante **returncode** : Code retour une fois le fichier de directive terminé.	Sous certains environnements (ex : WinPE), les demandes de redémarrage et d'arrêt ne sont pas pris en compte.

Value	Pour le type ***quit*** : **Code retour que l'application ExecAction retournera en sortie** Pour le type ***returncode*** : **Code retour quand l'application ExecAction aura terminé le fichier de directive normalement.**	: MEMO, EXEC. Il est possible de spécifier un code retour, mais attention il sera surcharger si une erreur survient avant la fin du fichier.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<SESSION type="logoff"/>

<SESSION type="quit" value="1"/>

<ACTION name="action_essai" descriptionAction="Action en essai" version="1">
      <MSGBOX title="Action avant" text="Le texte s'affiche" type="ok"/>

      <SESSION type="actionquit"/>

      <MSGBOX title="Action après" text="Le texte ne s'affiche pas !" type="ok"/>
</ACTION>


<SESSION type="returncode" value="#connexion_p#"/>
```

Créé avec HelpNDoc Personal Edition: Créer des documents d'aide PDF facilement

PAUSE

PAUSE

Description :

Effectue une pause dans le fichier de directive.

Type des attributs :

Obligatoire :

- **Value** : Valeur du temp de pause.

Optionnel (en fonction du contexte) :

- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Value	**y:xx** : Nombre d'années. **o:xx** : Nombre de mois. **w:xx** : Nombre de semaines. **d:xx** : Nombre de jours. **h:xx** : Nombre d'heures. **m:xx** : Nombre de minutes. **s:xx** : Nombre de secondes.	Les valeurs peuvent être cumulé, séparé par une virgule.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le

	hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<PAUSE type="delay" value="s:5"/>

<PAUSE type="delay" value="h:5"/>

<PAUSE type="delay" value="s:5000"/>
```

Créé avec HelpNDoc Personal Edition: Générateur de documentations PDF gratuit

INPUTBOX

INPUTBOX

Description :

Affiche une fenêtre permettant à l'utilisateur de pouvoir inscrire un texte.
Il sera récupéré et stocké pour une utilisation ultérieur.

Type des attributs :

Obligatoire :

- **Name** : Nom de l'étiquette qui mémorise la valeur.

Optionnel (en fonction du contexte) :

- **Title** : Titre de la fenêtre.
- **Message** : Texte qui sera une consigne ou une explication.
- **DefaultValue** : Valeur par défaut qui remplira la fenêtre.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Name	**Nom de l'étiquette de mémorisation.**	
Title	**Titre alphanumérique**	
Message	**Message alphanumérique**	
DefaultValue	**Valeur par défaut alphanumérique**	Option visible par l'utilisateur.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<INPUTBOX name="cheminuser" title="Super Démo NDF ExecAction" message="Voici une fenètre en InputBox" defaultvalue="C'est très fort"/>
```

Créé avec HelpNDoc Personal Edition: Produire des livres EPub gratuitement

FILEBOX

FILEBOX

Description :

Affiche une fenêtre permettant à l'utilisateur de pouvoir spécifier un fichier.
Il sera récupéré et stocké pour une utilisation ultérieur.

Type des attributs :

Obligatoire :

- **Name** : Nom de l'étiquette qui mémorise la valeur.

Optionnel (en fonction du contexte) :

- **Title** : Titre de la fenêtre.
- **FileFilter** : Filtre de fichier pour masquer certain type.
- **DefaultFile** : Valeur par défaut qui remplira la fenêtre.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?

Name	**Nom de l'étiquette de mémorisation.**	
Title	**Titre alphanumérique**	
FileFilter	**Description|extension** : Permet de masquer certaines extension non souhaité.	Par défaut tout les extensions seront visible. (Tout les fichiers|*.*
DefaultFile	**Valeur par défaut alphanumérique**	Option visible par l'utilisateur.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

Créé avec HelpNDoc Personal Edition: Générateur complet de livres électroniques ePub

PATHBOX

Description :

Affiche une fenêtre permettant à l'utilisateur de pouvoir spécifier un répertoire.
Il sera récupéré et stocké pour une utilisation ultérieur.

Type des attributs :

Obligatoire :

- **Name** : Nom de l'étiquette qui mémorise la valeur.

Optionnel (en fonction du contexte) :

- **Title** : Titre de la fenêtre.
- **DefaultPath** : Valeur par défaut qui remplira la fenêtre.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Name	**Nom de l'étiquette de mémorisation.**	
Title	**Titre alphanumérique**	
DefaultPath	**Emplacement par défaut (répertoire).**	Option visible par l'utilisateur.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

	-1 ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	

Exemple :

Créé avec HelpNDoc Personal Edition: Produire des livres électroniques facilement

HTTP

HTTP

Description :

Permet de télécharger un fichier.

Le système de proxy n'est pas prise en charge.

Type des attributs :

Obligatoire :

- **Url** : URL du fichier à télécharger.
- **LocalFile** : Chemin du fichier enregistré.

Optionnel (en fonction du contexte) :

- **Retry** : Nombre de tentative.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Url	**Chemin http, du fichier à télécharger.**	Dans le chemin donné, les caractères "*&*" son à transformer en "*<*". Sinon la chaîne ne sera pas compatible avec le format XML.
LocalFile	**Chemin du fichier enregistré.**	
Retry	**Nombre d'essais**	Si le téléchargement à échoué il est possible de recommencer un nombre défini de fois.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<HTTP url="http://www.ndfaciliti.com/toto.exe" localfile="d:\toto.exe" retry="2"/
>
```

Créé avec HelpNDoc Personal Edition: Générateur d'aides CHM gratuit

WAIT

WAIT

Description :

La balise WAIT permet de temporiser et de synchroniser le fichier de directive avec un événement extérieur.

Si la condition de l'attente n'est pas remplis alors il ne passera pas dans les sous directives.

Cette directive, mal utilisé peut entraîner le blocage du séquencement de vos opérations.

Type des attributs :

Obligatoire :

type : Détermine le type de structure a vérifier.

Optionnel (en fonction du contexte) :

mode : Configure le mode de vérification du type spécifié (type). Peut être obligatoire pour certain vérification.

location : Emplacement à vérifier ou chemin fichier ou emplacement et / ou clef dans le registre. Paramètre pouvant accepter les valeurs de type "expand".

blockTag : Condition de passage dans la balise check.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**file** : vérifie un fichier. **regkey** : Attend la présence d'un clef dans la base de registre. **regvalue** : vérifie dans la base de registre, la présence d'une valeur et de son contenu. **process** : vérifie un nom de processus en mémoire. **ping** : Attend la présence d'une machine	
Mode	Pour le type ***file*** : **detect** : Mode par défaut, teste juste la présence. **lost** : Si le fichier n'existe pas ou plus. Pour le type ***regkey*** : **detect** : Mode par défaut, teste juste la présence. **lost** : Si a clef n'existe pas ou plus. Pour le type ***regvalue*** : **detect** : Mode par défaut, test la valeur contenu. **lost** : Si la valeur spécifié n'est plus la même. Pour le type ***process*** : **detect** : mode par défaut, test si le processus existe. **lost** : Si le processus n'existe pas ou plus. Pour le type ***ping*** : **detect** : Tant que la machine n'est pas présente. **lost** : Si la machine n'est pas ou plus présente.	

Location	Pour le type ***registry*** : Emplacement de la clef ou de la valeur. Pour le type ***file*** : Chemin du fichier.	Les valeurs peuvent contenir des chaînes d'abstraction.
Value	Pour le type ***process*** : Nom du processus. Pour le type ***ping*** : Nom de la machine. Pour le type ***regvalue*** : Valeur de comparaison.	Pour le type ***process*** : Il est possible de spécifier le chemin complet d'un exécutable. Seul celui-ci sera strictement pris en compte lors de la vérification. Pour le type ***regvalue*** : Si la valeur est hexadécimale il faut spécifier la valeur tout en majuscule (ex: 00F34CE987)
TimeOut	Temps d'attente maximum en secondes.	
RefreshTime	Fréquence d'actualisation de l'opération par secondes.	Par défaut établie sur 1 seconde.
ValueName	Pour le type ***regvalue*** : Nom de la valeur.	
BlockTag	**0** ou **pie** : Passage si existe (par défaut). *pie : Pass If Exist* **1** ou **pne** : Passage si n'existe pas. *pne : Pass if Not Exist* **2** ou **signalerror** : Remonte en erreur si n'existe pas comme une n'importe quel autre directive.	Si l'attribut n'est pas spécifié est sera automatiquement généré par défaut.

Exemple :

Présence d'un fichier :

```
<WAIT type="file" mode="detect" location="d:\trig.txt" timeout="5">
    ...
    ...
</WAIT>
```

Ping d'une machine :

```
<WAIT type="ping" mode="lost" location="10.0.0.1" timeout="5">

    ...
    ...
</WAIT>
```

Perte du ping sur machine :

```
<WAIT type="ping" mode="detect" location="mamachine" timeout="5">
    ...
    ...
</WAIT>
```

Clef de base de registre :

```
<WAIT type="regkey" mode="detect" location="HKEY_CURRENT_USER\Software\ee
\Actionzz" timeout="5">
    ...
    ...
</WAIT>
```

Précence d'un processus :

```
<WAIT type="process" mode="lost" value="notepad.exe" timeout="5">

    ...
    ...
</WAIT>
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<DEMO>

       <EXEC programlocation="#windir#\notepad.exe" parameters="#temp#\wait.log"/
>
       <WAIT type="process" mode="lost" value="#windir#\notepad.exe">

              <MSGBOX title="This program is lost" text="Notepad is closed"/>
       </WAIT>

</DEMO>
```

Valeur de base de registre :

```
<WAIT type="regvalue" mode="detect" location="HKEY_CURRENT_USER\Software\ee
\Actionzz\" valuename="pouet" timeout="5">
    ...
    ...
</WAIT>
```

Créé avec HelpNDoc Personal Edition: Nouvelles et informations sur les outils de logiciels de création d'aide

LOG

LOG

Description :

Permet de régler et mettre en place un fichier journal contenant l'activité du fichier de directive.

Type des attributs :

Optionnel (en fonction du contexte) :

- **Location** : Emplacement + nom du fichier journal.
- **Mode** : .
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Location	**Emplacement du fichier journal**	
Type Cette option n'est plus disponible.	**LOG** : (par défaut) Créer un fichier à plat **HTML** : Format HTML **CSV** : Format CSV.	Cette option n'est plus disponible.
Mode	**Start** : (par défaut) Démarre l'enregistrement de fichier journal. **Pause** : Met l'enregistrement en pause. Il permet de toujours pouvoir enregistrer des informations avec la balise LOG, mais les actions et directives ne peuvent pas enregistrer. **Stop** : Arrête l'enregistrement de fichier journal.	Gère l'enregistrement des actions et des directives.
MaxSize	**Taille du fichier en Octet.**	Il est impératif de spécifier l'option *MaxArchive*.
MaxArchive	**Nombre de fichiers de rotation.**	Il faut avant tout spécifier une taille limite avec l'option *MaxSize*.
Text	**Texte a insérer dans le fichier de log**	
BlockTagError	**0 ou 1** ou **block** : Bloque sur l'erreur.	

	-1 ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	

Exemple :

```
<LOG location="d:\toto2.log" text = "coucou" MaxArchive="3" maxsize="256"/>
```

Créé avec HelpNDoc Personal Edition: Générer des livres électroniques EPub facilement

WHILE

WHILE

Description :

Boucle sur un ensembles de directive. Tant que la valeur existe ou est égale ou différente de ...

Sur chaque bouclage, la valeur, est stocké dans le mémo ***#_valueline_#***. Le nombre de passage est stocké dans ***#_linenumber_#*** . A Chaque passage ces valeurs sont modifiées.

Type des attributs :

Obligatoire :

Type : Type de service demandé.

Optionnel (en fonction du contexte) :

- **Value** : Valeur de retour.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**numeric** : Vérification par comparaison numérique. **file** : parcours un fichier spécifié ligne par ligne	
Mode	Pour le type ***numeric, file*** : **equal** (par défaut): égal à... **dif** : Valeur différente de ...	
Value	Pour le type ***numeric, file*** : **Code retour que l'application ExecAction retournera en sortie**	Il est possible de spécifier un code retour, mais attention il sera surcharger si une erreur survient avant la fin du fichier.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<DEMO>

       <MEMO type="internal" name="LocalDir" value="d:\testdir"/>

       <CREATE type="directory" location="#LocalDir#"/>

       <WINDOW textColor="#FFFFFF"
                     titleWindow="Test dir while"
                     backgroundColor="#385214"
                     textBackgroundListAction="#123456"
                     textColorListAction="#552233"
                     levelDisplay="rootAction"
       />


       <LOG location="#LocalDir#\activity.log" archivemax="3" maxsize="1024"/>


              <MEMO type="internal" Name="File" Value="1"/>

              <WHILE Type="numeric" Mode="dif" valuename="File" value="0">
                     <CHECK Type="file" Location="d:\test.txt">
                            <MEMO type="internal" Name="File" Value="0"/>
                            <PAUSE value="s:1"/>
                     </CHECK>
              </WHILE>
              <DELETE Type="file" Location="d:\test.txt" BlockTagError="ignore" /
>

</DEMO>
```

Créé avec HelpNDoc Personal Edition: Création d'aide CHM, PDF, DOC et HTML d'une même source

FOR

FOR

Description :

Boucle sur un ensembles de directive.

Sur chaque bouclage, la valeur, est stocké dans le mémo ***#_valueline_#***. Le nombre de passage est stocké dans ***#_linenumber_#*** . A Chaque passage ces valeurs sont modifiées.

Type des attributs :

Obligatoire :

- **Type** : Type de bouclage.

Optionnel (en fonction du contexte) :

- **Location** : Emplacement d'un fichier qui servira pour boucler.
- **Value** : Valeur de départ pour le bouclage.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**count** : Compteur de passage. **file** : parcours un fichier spécifié ligne par ligne. **filedir** :(option en cours de conception) parcours un répertoire fichier par fichier	
Location	**Emplacement d'un fichier**	Permet de spécifier un fichier, afin de de boucler sur chaque lignes. Les valeur sont récuperer dans le mémo #_valueline_#
Value	**Valeur de référence de compteur**	
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus	Aucune erreur ne sera remonté si l'application ne se lance pas correctement, quand l'appel est fait du coté interface.

	de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

Boucle FOR simple :

```
<FOR type="file" location="d:\log.txt">
   <MSGBOX title="Mon super titre" text="Coucou ! #_valueline_#" type="ok"/>
</FOR>
```

Boucle FOR imbriquées :

```
<FOR type="count" value="12">
          <MEMO type="internal" name="ValeurMajeur" value="#_valueline_#"/>
          <FOR type="count" value="12">
                 <MEMO type="internal" name="ValeurMineur"
value="#_valueline_#"/>
                 <MSGBOX title="compteur" text="#ValeurMajeur# #ValeurMineur#
#_valueline_#"/>
                 <HTTP url="http://www.monsite.com/#ValeurMajeur#-
#ValeurMineur#.pdf" localfile="d:\temp\#ValeurMajeur#-#ValeurMineur#.pdf"
blocktagerror="ignore"/>
          </FOR>
   </FOR>
```

Créé avec HelpNDoc Personal Edition: Générateur complet de livres électroniques ePub

LIB

LIB

Description :

Charge un *plugin* pour ExecAction. Cette option permet d'étendre ExecAction avec d'autre fonctionnalité. Une explication pour réaliser les plugin's est décrite ICI.

Type des attributs :

Obligatoire :

- **Name** : Nom permettant de spécifier la librairie.

Optionnel (en fonction du contexte) :

- **Type** : Charger / décharger la librairie.
- **Location** : Chemin du fichier librairie.
- **Function** : Nom de la fonction qui sera utilisé.
- **Name** : Sera utilisé pour vérifier ultérieurement le code retour avec un Check
- **pType[X]** : spécifie le type de paramètre a passer a la fonction. Le X indique le numéro a placer en fonction du paramètre.
- **pValue[X]** : spécifie la valeur de paramètre a passer a la fonction. Le X indique le numéro a placer en fonction du paramètre.
- **ValidReturnCode**: Code retour qui doit être validé après l'exécution de la fonction.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**UseFonc** (par défaut) : permet d'utiliser une fonction dans une librairie	Pour utiliser une fonction, il faut d'abord avoir chargé.

	Load : Charge une librairie en mémoire. **Close** : Décharge une librairie en mémoire.	
Location	**Chemin du fichier librairie**	Lors du chargement ExecAction vérifia la conformité de la librairie.
Function	**Nom de fonction**	ExecAction vérifiera si si la fonction existe bien. En revanche, aucune vérification sur le nombre de paramètre ne sera effectuée.
Name	**Nom de la directive** : Il sera possible de vérifier.	
ptype[X]	**String** : Défini que la valeur du paramètre X de la fonction sera de type texte. **integer** : Défini que la valeur du paramètre X de la fonction sera de type numérique. **double** : Défini que la valeur du paramètre X de la fonction sera de type décimale. **float** : Défini que la valeur du paramètre X de la fonction sera de type flottant. **quad** : Défini que la valeur du paramètre X de la fonction sera de type Integer de 8 octets.	ptype voulant dire parameter type, doit être précédé d'un numéro en fonction du numéro de paramètre de la fonction qui est appelé depuis la librairie.
pvalue[X]	**Valeur a passer au paramètre de la fonction**.	pvalue voulant dire parameter value, doit être précédé d'un numéro en fonction du numéro de paramètre de la fonction qui est appelé depuis la librairie.
ValidReturnCode	**Code retour** : Code retour validant la bonne exécution de la fonction appelée.	Il est possible de validé plusieurs code retour en séparant les valeurs avec **,** Si aucune cette option n'est pas spécifié aucun code retour ne sera prise en compte et l'exécution sera toujours validé.

BlockTagError	**0 ou 1** ou **block** : Bloque sur l'erreur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Règle la gestion des erreurs, pour les directives .

Exemple :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<DEMO_PLUGIN description="This is a demo for a plug"
locationRegistry="HKEY_CURRENT_USER\Software\plugexecaction">
       <LOG location="#temp#\demo_plugin.log" archivemax="3" maxsize="1024"/>

       <WINDOW textColor="#FFFFFF"
                     titleWindow="Super ESSAI"
                     logoImage="e:\ndf_execAction\logo.png"
                     backgroundColor="#385214"
                     textBackgroundListAction="#123456"
                     textColorListAction="#552233"
                     levelDisplay="rootAction"
       />

       <PLUGIN name="testdll" type="load" location="G:\Projets\ndf_execAction\v4
\Exemple\test.dll">
              <PLUGIN name="testdll" function="pluginMsg" ptype1="string"
pvalue1="Mon Titre" ptype2="string" pvalue2="Mon corp de message"
ptype3="integer" pvalue3="5" validReturnCode="3,6,1,5" />
       </PLUGIN>

</DEMO_PLUGIN>
```

Créé avec HelpNDoc Personal Edition: Générateur de documentation d'aide HTML gratuit

PLUGIN

PLUGIN

Description :

Charge un *plugin* pour ExecAction. Cette option permet d'étendre ExecAction avec d'autre fonctionnalités. Une explication pour réaliser les plugin's est décrite ICI.

Ce plugin une fois chargé, il sera donc possible d'utiliser le plugin, comme une balise dans le fichier de directive.

Lors du chargement plusieurs fonctionnalité peuvent être disponible en fonction du l'auteur.

Type des attributs :

Obligatoire :

- **Name** : Nom permettant d'utiliser le plugin comme une balise.

Optionnel (en fonction du contexte) :

- **RedirectionFS64** : Gère l'activation de la redirection du système de fichier x64.
- **WorkingDir** : Répertoire de travail d'ExecAction.
- **WaitAtClose** : Délai d'attente avant fermeture générale de ExecAction.
- **NoPluginInfo** : Masque les informations du plugin.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**Load** (par défaut) : permet d'utiliser une fonction dans une librairie **Close** : Décharge une librairie en mémoire.	

Location	**Chemin du plugin à charger**	Lors du chargement ExecAction vérifia la conformité de la librairie.
Name	**Nom de la balise**	le nom spécifié sera utilisé comme nom de balise, il sera donc possible dans le fichier de directive d'utiliser le plugin avec son nom de balise. Les balises d'ExecAction ne peuvent pas être surchargé, tout nom similaire à une balise déjà existante sera ignoré.
NoPluginInfo	**True ou 1** : Plus d'affichage des informations du plugin au chargement. **False ou 0** (par défaut) : Affichage des informations du plugin au chargement.	
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Règle la gestion des erreurs, pour les directives.

Exemple :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<DEMO_PLUGIN description="This is a demo for a plug"
locationRegistry="HKEY_CURRENT_USER\Software\plugexecaction">
       <LOG location="#temp#\demo_plugin.log" archivemax="3" maxsize="1024"/>

       <WINDOW textColor="#FFFFFF"
                     titleWindow="Super ESSAI"
                     logoImage="e:\ndf_execAction\logo.png"
                     backgroundColor="#385214"
                     textBackgroundListAction="#123456"
                     textColorListAction="#552233"
                     levelDisplay="rootAction"
       />

       <PLUGIN name="testdll" type="load" location="G:\Projets\ndf_execAction\v4
\Exemple\test.dll">
              <TESTDLL attribut_for_fonction="Hello world"/>
       </PLUGIN>

</DEMO_PLUGIN>
```

Créé avec HelpNDoc Personal Edition: Générateur complet d'aides multi-formats

SYSTRAY

SYSTRAY

Description :

Permet d'afficher un icône dans la barre de tâche et une infobulle.

Type des attributs :

Optionnel (en fonction du contexte) :

- **Name** : Nom permettant de spécifier une icône dans la barre des tâches.
- **Type** : Créer, supprimer, notifier. Indique le comportement de la directive.
- **BalloonIco** : Chemin ou nom générique d'un icône pour l'icône en

barre des tâches.

- **Title** : Titre du message (limité à 255 caractères).
- **Text** : Texte du message.
- **Description** : Texte quand la souris passe sur l'icône dans la barre des tâches.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**Create** (par défaut) : Créer un icône dans la barre des tâches. **Delete** : Enlève l'icône dans la barre des tâche. **Notify** : Notification utilisateur	IL n'y pas besoin de spécifier systématiquement des options pour le notifications. Par défaut le message sera affiché avec icône générique.
Icone	**Chemin du fichier icône** (par défaut) **app** : icône d'une application générique **asterisk** : "croix rouge" **error** : "Panneau attention" **warn** : "Point intérrogation" **info** : "logo information" **question** : "point d'intérrogation" **protec** : "bouclier de protection"	Si l'image indiqué n'existe pas, il affichera l'icône de l'application générique.
BallonIco	**None** (par défaut) : rien **Error** : "croix rouge"	

	Warn : "Panneau attention" **Info** : "Point intérrogation"	
Name	**Nom du l'occurrence de l'icône dans la barre des tâches.**	
Title	**Titre de l'infobulle**	255 caractères, maximum.
Text	**Message**	
Description	**Description de l'icône**	Sert que si on passe la souris sur l'icône dans la barre de tâches.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<DEMO log="#temp#\test_systray.log">
	<!-- <SYSTRAY title="My Title" text="Test for explain"/> -->
	<SYSTRAY/>
	<SYSTRAY Name="monIco" Icon="1329128009_application-xml.ico"/>
	<PAUSE Value="s:10"/>
	<SYSTRAY Type="notify" Name="monIco" title="Notify with a systray"
text="Hello world !" TimeOut="3"/>
	<PAUSE Value="s:10"/>
```

```
</DEMO>
```

Créé avec HelpNDoc Personal Edition: Création d'aide CHM, PDF, DOC et HTML d'une même source

EXPORT

EXPORT

Description :

Exporte ou sauvegarde les informations utilisé par le logiciel lors de son fonctionnement.
Il permet d'exporter des informations interne, pour être éventuellement réutilisé par la suite.

Type des attributs :

Obligatoire :

Type : Sélection des informations a exporter (mémo, box, etc...).

Mode : Choix dans le format ou la catégorie d'export (base de registre, fichier,etc...).

Optionnel (en fonction du contexte) :

Location : Chemin pour l'exportation en fonction du mode choisi

BlockTagError : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**Memo** : export des mémo. **Exec** : export des retours	Il est impératif de spécifier un type d'export.

	d'exécution. **Box** : Export des retours des différentes fenêtres de dialogue. **Cmdlinesparameters** : Exporte les différents paramètre récupéré de la ligne de commande passé lors du lancement de ExecAction.	
Mode	**registry** : Chemin d'un clef dans la base de registre **csv** : Chemin d'un fichier, qui sera au format CSV **ini** : Chemin d'un fichier, qui sera au format INI	Cette option est valable pour chaque Type
Location	**Emplacement du fichier ou de la clef de base de registre où seront contenu les valeurs.**	
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Règle la gestion des erreurs, pour les directives .

Exemple :

Créé avec HelpNDoc Personal Edition: Créer des documents d'aide PDF facilement

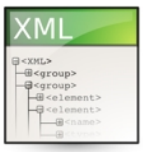
Spécifique

Balises spécifiques

Ici sont décrites les balises qui sont utilisé pour exécuter des actions.

Créé avec HelpNDoc Personal Edition: Générer des livres électroniques EPub facilement

Racine action



Racine action

Description :

Balise obligatoire. Elle permet de définir des réglages sur le contexte d'exécution des directives et des actions.

Elle permet aussi de modifier le comportement de l'utilitaire ExecAction.

Cette balise n'a pas de nom propre, mais il est *obligatoire* de mettre un nom de balise.

Type des attributs :

Optionnel (en fonction du contexte) :

- **Description** : Rapide description qui sera inséré dans les journaux exécutions et à l'affichage.
- **TitleWindow** : Titre de la fenêtre dans la cas de l'affichage graphique.
- **LocationRegistry** : Emplacement dans la base de registre pour stocker les acquittements d'actions (version et date).
- **EnableRegistry64** : Active ou désactive la redirection 64btis dans la base de registre.
- **ActionFilePath** : Chemin du fichier.
- **LoadUserGroupAD** : Charge les groupes dont l'utilisateur fait partis.
- **Log** : Chemin et nom du fichier journal.
- **BlockAction** : Configure le comportement de la gestion des erreurs par défaut des actions.
- **BlockTagError** : Configure le comportement de la gestion des erreurs par défaut des directives.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Description	**Nom de l'action**	Chiffre et lettre seulement.
LocationRegistry	**Chemin dans la base de registre**	Si l'arborescence n'est pas créer, il le sera automatiquement.
EnableRegistry64	**False** (par défaut) : Enlève la redirection 64Bits de la base de registre **True** : Active la redirection 64bits de la base de registre	Par défaut la redirection est automatiquement désactivé.
ActionFilePath		

	Chemin du fichier des actions	Si le fichier n'est pas crée il le sera automatiquement. Cette option indique de l'utiliser avec la version d'ExecAction permettant l'enregistrement des actions dans un fichier.
LoadUserGroupAD	**0** : Ne charge pas les groupes utilisateur. **1** : Charge pas les groupes utilisateur.	Charge les groupes auquel l'utilisateur fait partie.
Log	**Emplacement du fichier journal**	Cette option permet juste d'indiquer l'emplacement, la directive <LOG/> pourras reconfigurer les options par la suite. Cela permet de loguer les actions de la racine, avant d'avoir réellement défini un fichier journal.
BlockTagError	**0 ou 1** ou **block** : Bloque sur l'erreur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Règle la gestion des erreurs, pour les directives.
BlockAction	**0 ou 1** ou **block** : Bloque sur l'erreur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Règle la gestion des erreurs, pour les actions.

Exemple :

```
<CONFIG_MACHINE titleWindow="Super Script" description="Configuration
Machine" log="#temp#\essai.log" loadUserGroupAD="1" displayMode="console"
```

```
levelDisplay="all" locationRegistry="HKEY_CURRENT_USER\Software\ND-Faciliti
\config_machine" blockTagError="-1">
       ...
       ...
       </CONFIG_MACHINE>
```

```
       <CONFIG_MACHINE textColor="FFFFFF" backgroundColor="#0000FF" logoImage="d:
\temp\ndf2.ico" log="#temp#\essai_3.log" loadUserGroupAD="1" displayMode="window"
levelDisplay="rootAction" locationRegistry="HKEY_CURRENT_USER\Software\ND-
Faciliti\essai" blockTagError="0">
       ...
       ...
       </CONFIG_MACHINE>
```

```
       <DEMO_NDF_EXECACTION displayMode="window" titleWindow="Super ESSAI"
logoImage="F:\ndf_execAction\logo.png" backgroundColor="#385214"
textBackgroundListAction="#123456" textColorListAction="#552233"
textColor="#FFFFFF" description="Configuration Machine" log="#temp#
\DEMO_NDF_EXECACTION.log" loadUserGroupAD="1" levelDisplay="all"
locationRegistry="HKEY_CURRENT_USER\StrategieNdf\DEMO_ACTION">
       ...
       ...
       </DEMO_NDF_EXECACTION>
```

```
   <DEMO_NDF_EXECACTION displayMode="window" titleWindow="Super ESSAI"
   logoImage="F:\Protjets\ndf_execAction\logo.png" backgroundColor="#385214"
   textBackgroundListAction="#123456" textColorListAction="#552233"
   textColor="#FFFFFF" description="Configuration Machine" log="#temp#
   \DEMO_NDF_EXECACTION.log" loadUserGroupAD="1" levelDisplay="rootAction"
   locationRegistry="HKEY_CURRENT_USER\StrategieNdf\DEMO_ACTION">
       ...
       ...
       </DEMO_NDF_EXECACTION>
```

Créé avec HelpNDoc Personal Edition: Créer des documentations web iPhone

ACTION

ACTION

Description :

Regroupe un ensemble de directives permettant le contrôle sur l'exécution.

Il est possible de déclencher les directives contenu dans les actions sur une version de l'action, une date, une période.

Type des attributs :

Obligatoire :

Name : Nom de l'action. Cette valeur sera aussi inscrit dans la base de registre pour numéroter les versions ou les dates.

Optionnel (en fonction du contexte) :

Version : Version de l'action. C'est une condition de passage.

VersionDate : Version par période chronique. C'est une condition de passage.

DescriptionAction : Rapide description qui sera inséré dans les journaux d'exécutions.

BlockAction : Gestion des erreur des sous balise. C'est une condition de validation.

BlockTagError : Gestion des erreur des sous balise. C'est une condition de validation.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Name	**Nom de l'action**	Chiffre et lettre seulement.
Version	**0** (par défaut) : Passage dans l'action systématique. **-1** : L'action est désactivé. **1 à 21447483647** : Passage dans le	Si la condition de passage est validé, alors les directives contenu dans le groupe, seront exécutés. Si la version est supérieur dans le fichier de direction, par rapport a ce qui est contenu dans la base de

	groupe, en fonction de la différence entre l'inscription sur la machine et le numéro de version.	registre, alors l'exécution des directives sera lancé. Il est possible de cumuler les deux attributs Version et VersionDate.
VersionDate	**y:xx** : Nombre d'années **o:xx** : Nombre de mois **w:x** : Nombre de semaines **d:xx** : Nombre de jours **h:xx** : Nombre d'heures **m:xx** : Nombre de minutes **s:xx** : Nombre de secondes	Par défaut aucune option n'est spécifié. Il faut donc spécifier systématiquement des valeurs. Il est possible de spécifier plusieurs valeurs, le caractère séparateur des différentes valeurs est le ;
DescriptionAction	**Commentaires alphanumérique**	Ces commentaires se retrouveront dans les journaux d'exécution.
BlockAction	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le paramètre de la racine action. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifiée, alors elle ne sera pas marqué comme effectuée.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Règle la gestion des erreurs, pour les directives.

Exemple :

Action simple :

```
	<ACTION name="action_essai" descriptionAction="Action en essai"
version="1" blockAction="block">

	</ACTION>
```

Actions combinées avec une vérification (balise CHECK) :

```
	<ACTION name="Vérification du XML" descriptionAction="Essai du XML"
version="0">
		<CHECK type="file" mode="md5" location="#temp#\essai.xml"
value="dsdf6s51sdfsd5651sfd">
			<ACTION name="Execution_msi"
descriptionAction="Essai MSI" version="1">
				<EXEC programLocation="#windir#\system32
\msiexec.exe" parameters="/i \\Monserveur\apps\monpackage.msi /qn /norestart"
mode="wait" waitingTime="5"/>
			</ACTION>
		</CHECK>
	</ACTION>
```

Actions combinées :

```
	<ACTION name="Vérification du XML" descriptionAction="Essai du XML"
version="0">
		<ACTION name="Execution_msi" descriptionAction="Essai MSI"
version="1">
		<EXEC programLocation="#windir#\system32\notepad.exe"
parameters="/?" mode="wait" waitingTime="5"/>
		</ACTION>
	</ACTION>
```

Action avec une période de fonctionnement (versionDate):

```
   <ACTION name="action_essai" descriptionAction="Action en essai" version="0"
   versionDate="d:1,h:3,m:5" blockAction="inherit">
      ...
```

```
    ...
</ACTION>
```

Créé avec HelpNDoc Personal Edition: Générateur d'aide complet

COMOGUI

COMOGUI

Description :

Configure une commande qui pourra être appelé depuis l'interface graphique.

Cette directive n'interfère en rien le déroulement du fichier de directive.

Type des attributs :

Obligatoire :

- **Name** : URL du fichier à télécharger.
- **ProgramLocation** : Chemin du fichier enregistré.

Optionnel (en fonction du contexte) :

- **Parameters** : Paramètre(s) qui sera passer à l'exécutable spécifié.
- **WorkingDirectory** : Répertoire de travail de l'exécutable.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Name	Nom de la commande qui sera disponible dans l'interface.	
ProgramLocation	Chemin du fichier exécutable	
Parameters	Paramètre qui sera fournis à l'exécutable	
WorkingDirectory	Répertoire de travail de l'exécutable	
BlockTagError	0 ou inherit : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. 1 ou block : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. -1 ou ignore : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Aucune erreur ne sera remonté si l'application ne se lance pas correctement, quand l'appel est fait du coté interface. Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

Exemple :

```
<COMOGUI name="cmd" ProgramLocation="#windir#\system32\cmd.exe" />
```

Créé avec HelpNDoc Personal Edition: Créer des documentations web iPhone

WINDOW

WINDOW

Description :

Configure une commande qui pourra être appelé depuis l'interface graphique.

Cette directive n'interfère en rien le déroulement du fichier de directive.

Type des attributs :

Obligatoire :

- **Name** : URL du fichier à télécharger.
- **ProgramLocation** : Chemin du fichier enregistré.

Optionnel (en fonction du contexte) :

- **Type** : Indique l'action a mener sur la fenêtre (création, modification, suppression).
- **Description** : Permet d'indiquer une description sur la fenêtre.
- **TitleWindow** : Titre de la fenêtre.
- **LevelDisplay** : Affiche les actions, directives simples, ou rien. Cela permet de régler le niveau de l'affichage.
- **TextColor** : Couleur du texte.
- **TextColorListAction** : Couleur du texte pour les listes des actions.
- **BackgroundColor** : Couleur de fond pour de la fenêtre.
- **BackgroundColorListAction** : Couleur de fond pour la liste des actions.
- **Icon** : Icône de la fenêtre.
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
Type	**Create** (par défaut) : Création	

	de la fenêtre. **Alter** : *option pas disponible.* **Hide** : Cache la fenêtre **Show** : Affiche la fenêtre caché. **Delete** : Supprime la fenêtre.	
Description	**Description**	
TitleWindow	**Titre de la fenêtre**	
DisplayLevel	**AllAction** (par défaut) : Affiche toutes les actions même celles déjà acquittées. **Action** : Affiche toutes les actions même *sauf* celles déjà acquittées. **All** : Affiche toute l'activité des actions et des directives.	
TextColor	Pour le type ***window*** : **#xxxxxx** ou **xxxxxx**	Les valeurs sont hexadécimal.
BackgroundColor	Pour le type ***window*** : **#xxxxxx** ou **xxxxxx**	Les valeurs sont hexadécimal.
TextColorListAction	Pour le type ***window*** : **#xxxxxx** ou **xxxxxx**	Les valeurs sont hexadécimal.
BackgroundColorListAction	Pour le type ***window*** : **#xxxxxx** ou **xxxxxx**	Les valeurs sont hexadécimal.
Icon	**Chemin de l'icône.**	Le format peut être : GIF,ICO,PNG,JPG. Attention a ne pas surcharger le poids pour un simple icône
LogoImage	**Chemin de l'image.** Insertion d'une image : (bmp,jpg,png,tif).	La largeur et hauteur de la fenêtre seront redéfini automatiquement en fonction de celle de l'image. Il est possible de spécifier une chaîne d'abstraction.
OpenListAction	**false :** (par défaut) fenêtre en mode résumé.	

	true : ouvre la fenêtre en mode détaillé.	
ForegroundWindow	**false :** (par défaut) fenêtre en mode normal. **true :** ouvre la fenêtre en mode premier plan.	Lorsque ExecAction lance un programme, externe en mode visible, alors la fenêtre se lance devant celle du parent. Avec cette option la fenêtre parent restera devant (au premier plan).
WindowX	Modifie la position de la fenêtre. Par défaut centré	
WindowY	Modifie la position de la fenêtre. Par défaut centré	
Log	**Emplacement du fichier journal**	
BlockTagError	**0 ou 1** ou **block** : Bloque sur l'erreur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris en compte.	Règle la gestion des erreurs, lors d'une erreur survenue sur la directive.

Exemple :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<DEMO_NDF_EXECACTION description="Configuration Machine" log="#temp#
\DEMO_NDF_EXECACTION.log" loadUserGroupAD="1" locationRegistry="HKEY_CURRENT_USER
\StrategieNdf\DEMO_ACTION">

	<WINDOW textColor="#FFFFFF"
			titleWindow="Super ESSAI"
			logoImage="e:\ndf_execAction\logo.png"
			backgroundColor="#385214"
			textBackgroundListAction="#123456"
			textColorListAction="#552233"
			levelDisplay="rootAction"
			Icon="c:\MyIcon.ico"
	/>

	<ACTION name="Execution_msi" descriptionAction="Essai MSI" version="0">

		<MSGBOX title="Mon super titre" text="Coucou !" type="ok"/>
	</ACTION>
```

```
</DEMO_NDF_EXECACTION>
```

Créé avec HelpNDoc Personal Edition: Produire facilement des livres électroniques Kindle

EXECACTION

EXECACTION

Description :

Modification du comportement du cœur de l'application ExecAction.

Type des attributs :

Optionnel (en fonction du contexte) :

- **RedirectionFS64** : Gère l'activation de la redirection du système de fichier x64.
- **EnableRegistry64** : Gère la redirection registre 32 Bits pour un environnement 64.
- **WorkingDir** : Répertoire de travail d'ExecAction.
- **WaitAtClose** : Délai d'attente avant fermeture générale de ExecAction.
- **LogComment** : Ajoute au fichier journal, les commentaires laissé dans le fichier de directive
- **BlockTagError** : Gestion des erreurs de la directive.

Description des attributs :

Attributs	Options	Comment l'utiliser ?
RedirectionFS64		Dans les cas où il est utilisé la

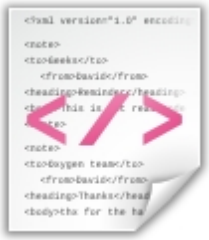
	False (par défaut) : La redirection FS64 est désactivé sous Windows. **True** : La redirection FS64 est activé sous Windows.	redirection ne s'appliquent pas a tout les dossiers, mais seulement au dossier système de Windows par exemple.
EnableRegistry64	**False** (par défaut) : La redirection de la base de registre dans le SysWow64 est désactivé sous Windows. **True** : La redirection est activé sous Windows.	Dans la base de registre, si un processus 32bits est exécuté il sera automatiquement redirigé dans la clef SysWow64.
LogComment	**False** (par défaut) : Aucun commentaires ne sera ajouter dans le fichier journal. **True** : Les commentaires se trouvant dans le fichier de directive seront automatiquement ajouté en lieu et place dans le fichier journal.	
WorkingDir	**Répertoire de travail**	Le changement de répertoire de travail ne s'applique qu'a ExecAction.
WaitAtClose	**Valeur d'attente en secondes**	A la fin de l'exécution du fichier de directive, il est possible de spécifier un temps de pause avant que l'application ne se termine.
BlockTagError	**0** ou **inherit** : Héritage (par défaut) : La balise reprendra le BlockTagError de la balise hiérarchiquement au dessus de lui. On parlera alors d'héritage. **1** ou **block** : Bloque en cas d'erreur : Cette action remontra en erreur au niveau supérieur. **-1** ou **ignore** : Ne bloque pas en cas d'erreur : Le retour ne sera pas pris	Indique si l'erreur doit être prise en compte par l'action. Il n'est pas nécessaire de le spécifier cet attribut. En effet si l'option n'est pas spécifié, alors elle sera automatiquement en mode héritage.

	en compte.	

Exemple :

Créé avec HelpNDoc Personal Edition: Écrire des livres électroniques ePub pour l'iPad

Fichier de directive XML



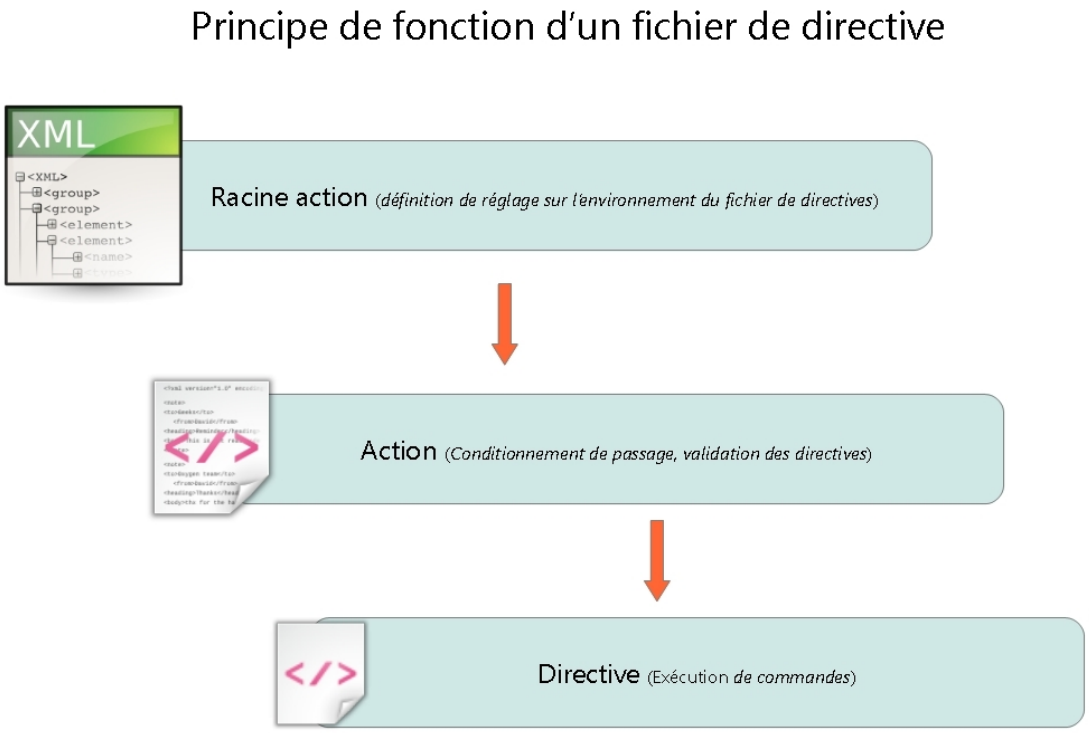
Fonctionnement du fichier de directive XML

Pour créer un fichier de directive il suffit tout simplement d'ouvrir un éditeur de texte et d'utiliser le format XML.

Pour plus de détail sur le format XML.

Le fichier de directive est au format XML. Comme tout format XML, il est possible de spécifier un type d'encodage.
par exemple

.....



Créé avec HelpNDoc Personal Edition: Créer des aides HTML, DOC, PDF et des manuels depuis une même source

Légende des icônes

Légendes et explication des icônes

En mode fenêtré (window), les actions sont affichées avec des icônes spécifiques pour chaque status de l'action.
En voici l'explication.

Légende :

Icône	Status	Description
	Exécution en cours.	Indique que l'action est en cours d'exécution. Ce statuts est éphémère car une fois terminé elle changera d'icône.
	Erreur sur l'action.	Une erreur est survenu sur cette action. Elle ne sera pas validé.
	Action déjà effectuée ou ignorée.	

		Cet icône sera affiché dans les cas ou l'action ou la sous action est ignoré car l'action est déja effectué. Mais aussi si la balise check est seule ou en dernier dans la liste de l'action.
	Action désactivée.	L'action est désactivé c'est à dire à -1 par exemple.
	Action terminée correctement.	Indique que l'action est arrivé a terme sans erreur. Attention toutefois si vous avez utilisez des surcharges d'erreur avec la fonction *blockAction*.
	Affichage d'événements.	Cet icône s'affiche si vous avec activé le mode "All". Cela permet ainsi de voir l'étendu de l'activité du fichier de directive.

Créé avec HelpNDoc Personal Edition: Création d'aide CHM, PDF, DOC et HTML d'une même source

Utilisation de l'executable

Utilisation de l'exécutable ExecAction

Dans un souci de contrainte de poids, deux version d'ExecAction sont fournis. Une avec la gestion des actions en fichier ou base de registre. L'autre sans la gestion des actions en mode fichier.

Le format des fichiers pour la gestion des actions est en SQLite. Comme dans la base de registre

Le comportement d'ExecAction entre les deux versions ne change pas.

Voici les différents paramètre pour se servir de l'exécutable ndf_execAction.exe :

Syntaxe :

/s => Silencieux.
/f => Chemin du fichier à traiter.
/j => Chemin du fichier journal pour le moteur d'execAction.
/t => Type de lecture XML, (ascii,utf8,unicode).
/m => Chemin d'un fichier mémo au format INI. Cela permettra de charger des mémo's avant tout fichier de directive.

Le format d'enregistrement des mémo sera du format #GROUPE_NOMVALEUR#, le fichier sera chargé complètement.

Valeur de retour :

0 => Aucune erreur, l'exécution s'est bien déroulé.
-1000 => Le fichier de directive n'est pas spécifié.
-998 => Impossible d'ouvrir le fichier spécifié.
-996 => Une erreur se trouve dans le fichier de directive, il a été impossible de l'interpréter.
-994 => Erreur sur l'exécution d'une balise.
-992 => Erreur sur l'acquisition d'un attribut.
-990 => Erreur sur l'ouverture du fichier mémo spécifié.

Exemple :

```
ndf_execAction.exe /f "essai.xml" /j "#temp#\ndf_execAction.log"
```

```
ndf_execAction.exe /f "essai_2.xml" /j "#temp#
\ndf_execAction.log" /s
```

Créé avec HelpNDoc Personal Edition: Générateur gratuit de livres électroniques et documentation

Exemple de scripts

Exemples de scripts

Exemple 1 :

```
	<?xml version="1.0" encoding="ISO-8859-1" ?>

<DEMO_NDF_EXECACTION displayMode="window" titleWindow="Super ESSAI"
description="Configuration Machine" log="#temp#\DEMO_NDF_EXECACTION.log"
loadUserGroupAD="1" modeAffichage="window" levelDisplay="All"
locationRegistry="HKEY_CURRENT_USER\Software\ND-Faciliti\DEMO_NDF_EXECACTION">
	<ACTION name="MESSAGEBX" version="0" descriptionAction="C'est une démo de
l'affichage d'une fenêtre">
		<MSGBOX title="Super Démo NDF ExecAction" text="Voici une fenêtre
en messsageBox" type="ok"/>
	</ACTION>

	<ACTION name="CREATION_RACCOURCIS" version="1" descriptionAction="Crée un
raccourcis sur C:\">
		<MSGBOX title="Création d'un raccourcis" text="Celui ci va créer
une fenêtre" type="ok"/>
			<SHORTCUT location="c:\essai.lnk" targetPath="#windir#
\notepad.exe" comment="Essai de creation de raccourcis" workingDirectory="%windir
%" openMode="maxi" fileIcon="#windir#\twunk_16.exe" fileIconIndex="2"/>
	</ACTION>
</DEMO_NDF_EXECACTION>
```

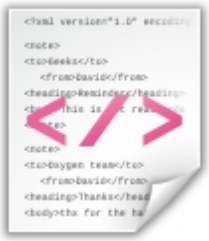
Exemple 2 :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<DEMO_NDF_EXECACTION displayMode="window" titleWindow="Super ESSAI" logoImage="F:
\Protjets\ndf_execAction\logo.png" backgroundColor="#385214"
textBackgroundListAction="#123456" textColorListAction="#552233"
textColor="#FFFFFF" description="Configuration Machine" log="#temp#
\DEMO_NDF_EXECACTION.log" loadUserGroupAD="1" levelDisplay="rootAction"
locationRegistry="HKEY_CURRENT_USER\StrategieNdf\DEMO_ACTION">
	<ACTION name="MESSAGEBX" version="0" descriptionAction="C'est une démo de
l'affichage d'une fenêtre">
		<MSGBOX title="Super Démo NDF ExecAction" text="Voici une fenêtre
en messsageBox" type="ok"/>
	</ACTION>

	<ACTION name="CREATION_RACCOURCIS" version="0" descriptionAction="Crée un
raccourcis sur C:\">
		<MSGBOX title="Création d'un raccourcis" text="Celui ci va créer
une fenêtre" type="ok"/>
			<SHORTCUT location="c:\essai.lnk" targetPath="#windir#
\notepad.exe" comment="Essai de creation de raccourcis" workingDirectory="%windir
%" openMode="maxi" fileIcon="#windir#\twunk_16.exe" fileIconIndex="2"/>
	</ACTION>

	<ACTION name="REGEXPAND" version="0" descriptionAction="Essai en expand">
		<REGISTRY mode="add" type="reg_sz" location="HKEY_CURRENT_USER
```

```
\StrategieNDF" key="essai_dans_regsitre" value="#j#/#m#/#a# - #h#:#m#:#s# -
#username#" blockTagError="-1"/>
        </ACTION>

        <ACTION name="COPY" version="0" descriptionAction="Essai en expand">
                <COPY type="file" sourceLocation="F:\Projets\ndf_execAction
\essai.log" targetLocation="d:\temp\azertyu\iop\qsd\fghj\toto.txt"
mode="createDir"/>
        </ACTION>
        <ACTION name="connexion_lecteurs" descriptionAction="Connexion des
lecteurs réseau" version="0">
                <EXEC programLocation="#windir#\system32\net.exe" parameters="use *
/d" mode="hidden,wait" waitingTime="10"/>
                <EXEC programLocation="#windir#\system32\net.exe" parameters="use
P: \\ndffl01\photos" mode="hidden,wait" waitingTime="10"/>
                <EXEC programLocation="#windir#\system32\net.exe" parameters="use
T: \\ndffl01\doncom" mode="hidden,wait" waitingTime="10"/>
                <EXEC programLocation="#windir#\system32\net.exe" parameters="use
R: \\ndffl01\dongro" mode="hidden,wait" waitingTime="10"/>
                <EXEC programLocation="#windir#\system32\net.exe" parameters="use
K: \\ndffl01\dupsil" mode="hidden,wait" waitingTime="10"/>
                <EXEC programLocation="#windir#\system32\net.exe" parameters="use
U: \\ndffl01\donuti\#username#" mode="hidden,wait" waitingTime="10"/>
                <EXEC programLocation="#windir#\system32\net.exe" parameters="use
L: \\ndffl01\LOGS\utilisateurs\#USERNAME#" mode="hidden,wait" waitingTime="10"/>
        </ACTION>
        <ACTION name="Execution_msi" descriptionAction="Connexion des lecteurs
réseau" version="0">
                <EXEC programLocation="#windir#\system32\msiexec.exe"
parameters="/?" mode="wait" waitingTime="5"/>
        </ACTION>
</DEMO_NDF_EXECACTION>
```

Créé avec HelpNDoc Personal Edition: Créer des documents d'aide facilement

Gestion des erreurs



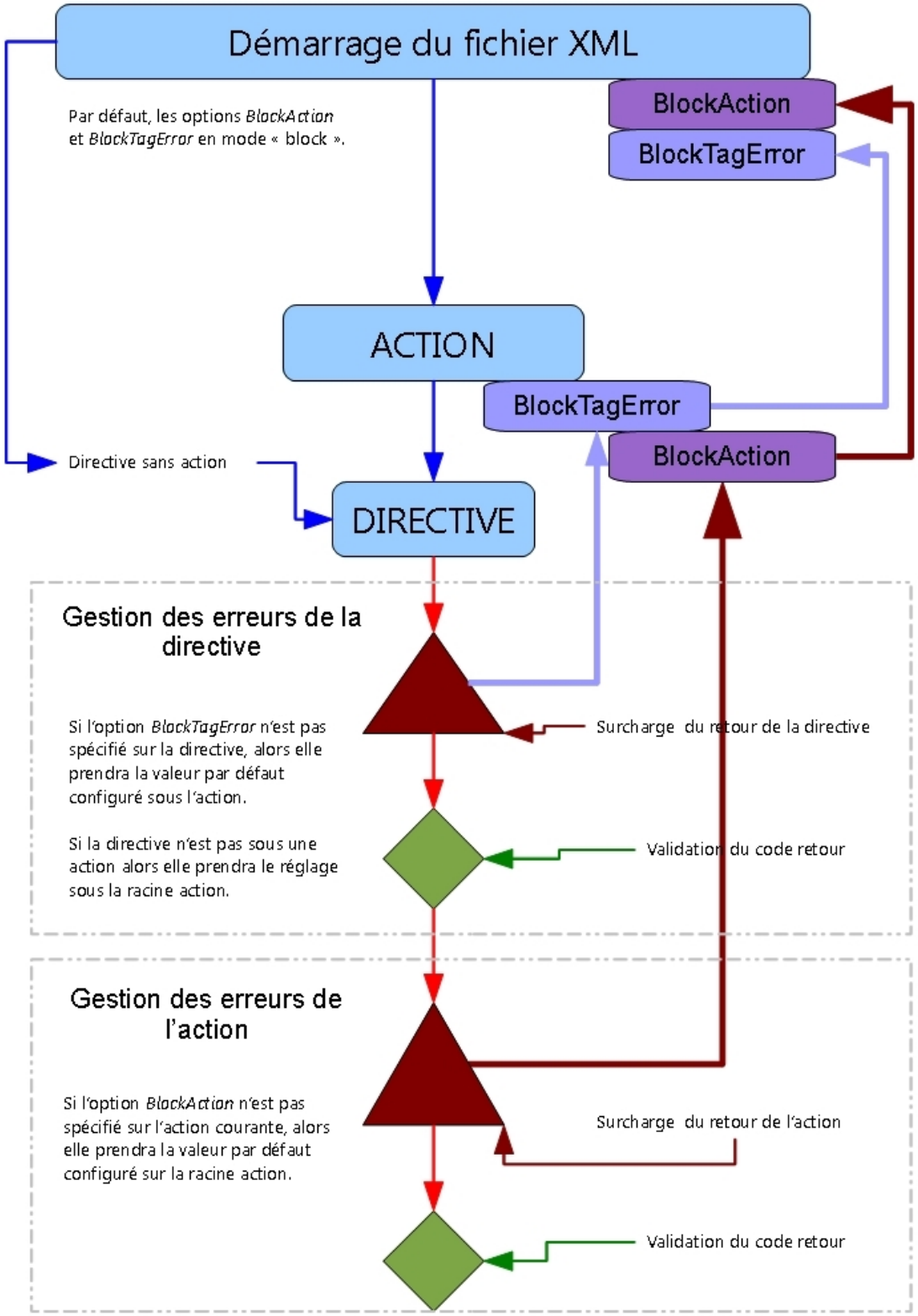
Gestion des erreurs

Les erreurs sont gérées par héritage successifs.
Si aucun option n'est défini, alors systématiquement dès qu'une erreur survient dans une directive, ExecAction sortira automatiquement.

Il est possible d'utiliser les erreurs par inversion. En effet il est possible de spécifier, *BlockAction* et *BlockTagError* en mode "*ignore*", puis de forcer une sortie obligatoire,

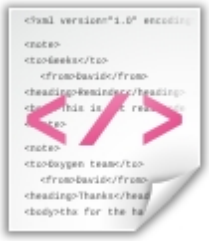
en spécifiant *BlockTagError* et *BlockAction* en mode "*block*".

.....



Créé avec HelpNDoc Personal Edition: Écrire des livres électroniques ePub pour l'iPad

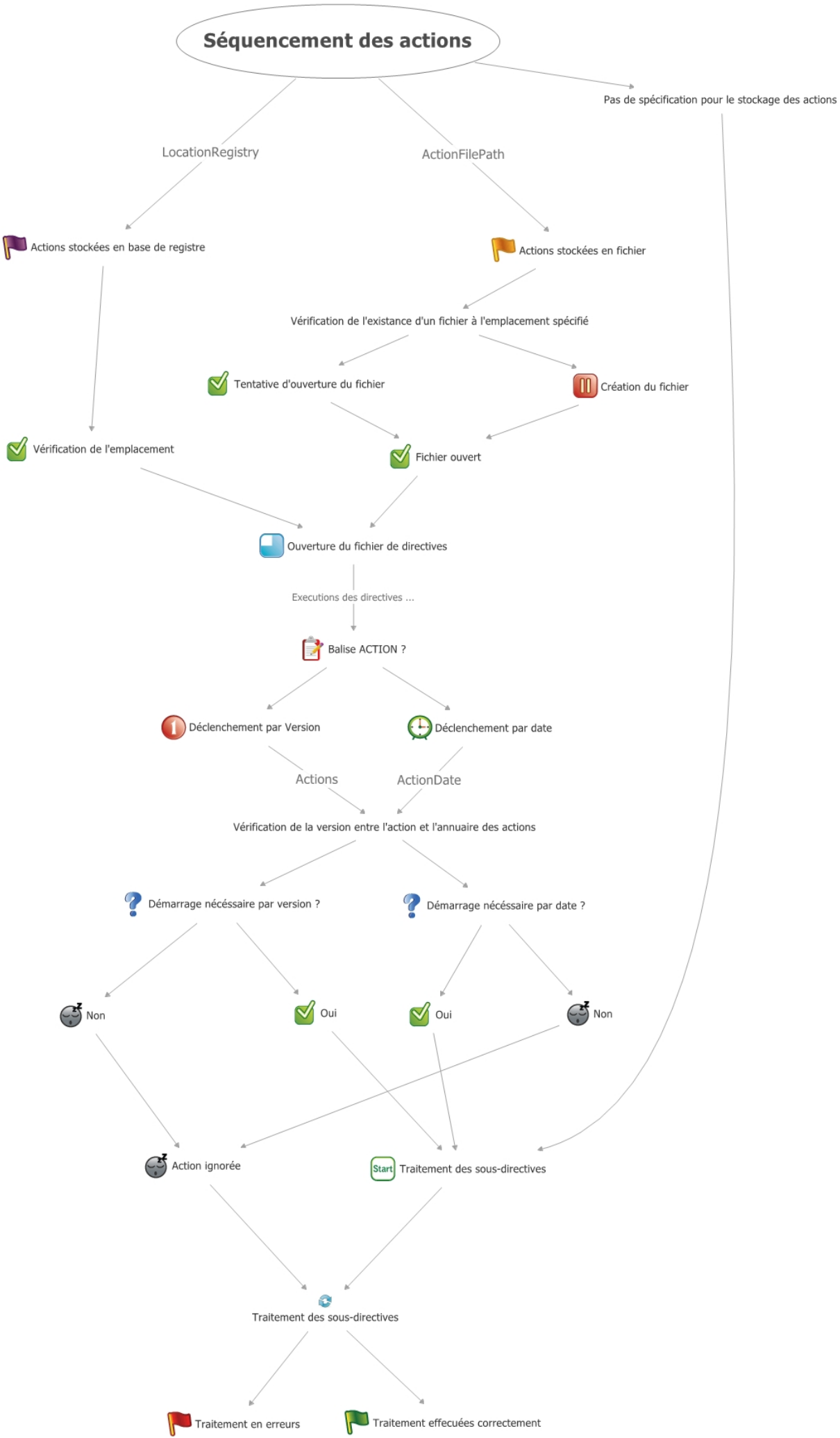
Déclenchement des actions



Déclenchement des actions

Cette partie décrit les étapes pour le démarrage ou non des actions à version numérique ou datées.

.....


Séquencement des actions
Pas de spécification pour le stockage des actions
LocationRegistry
ActionFilePath
Actions stockées en base de registre
Actions stockées en fichier
Vérification de l'existance d'un fichier à l'emplacement spécifié
Tentative d'ouverture du fichier
Création du fichier
Vérification de l'emplacement
Fichier ouvert
Ouverture du fichier de directives
Executions des directives ...
Balise ACTION ?
Déclenchement par Version
Déclenchement par date
Actions
ActionDate
Vérification de la version entre l'action et l'annuaire des actions
Démarrage nécéssaire par version ?
Démarrage nécéssaire par date ?
Non
Oui
Oui
Non
Action ignorée
Start
Traitement des sous-directives
Traitement des sous-directives
Traitement en erreurs
Traitement effecuées correctement

Créé avec HelpNDoc Personal Edition: Nouvelles et informations sur les outils de logiciels de création d'aide

Mémo constantes

Constantes Mémo et syntaxe en mode "expand"

Description :

Il est possibles de mettre des variables d'environnement ou bien des variables spécifique de ExecAction.
Il faut toujours les préfixer et les suffixer de "#".

Exemple de syntaxe :

<BALISE Value="#_xmlfile_#"/>

Certains "mémo" sont automatiquement créés dès qu'un fichier de directive démarre.
Ces mémo, sont utilisables comme n'importe qu'elle autre mémo. Il est même possible de modifier ces constantes. Pour les différencier, il sont préfixés et suffixés du caractère "_".

Bien respecter la case majuscule/minuscule pour les variables, notamment celles récupéré au système.

Mémo et son contenu :

Variable	Description	Options
h	Affiche les heures au format 24h.	
mi	Affiche les minutes.	
s	Affiche les secondes.	
yyyy	Affiche l'année.	
y	Année courant sur un format de deux chiffre	
m	Affiche le mois.	
d	Affiche le jours.	
currentdir	Emplacement du répertoire courant.	

programname	Le nom du programme (execaction).	
home	Chemin du répertoire de l'utilisateur courant	
xmlfile	Nom du fichier XML de directive utilisé.	Dès le fichier ouvert, cette variable est provisionné.
xmlfilepath	Emplacement absolu du fichier de directive XML.	Dès le fichier ouvert, cette variable est provisionné.
timestamp	Timestamp à l'heure courante	
action.name	Nom de l'action	Ne fonctionne que si une action est courante
action.version	Version de l'action	Ne fonctionne que si une action est courante
action.description	Description de l'action	Ne fonctionne que si une action est courante

Exemple :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<DEMO>
       <MSGBOX title="This is a demo" text="The memo constant _xmlfilepath_ is
#_xmlfilepath_#" type="ok"/>
       <MSGBOX title="This is a demo" text="The memo constant _xmlfile_ is
#_xmlfile_#" type="ok"/>
<REGISTRY mode="add" type="reg_sz" location="HKEY_CURRENT_USER\Software\ND-
Faciliti" key="essai_expand" value="#j#/#m#/#a# - #h#:#m#:#s#"/>

<COPIE type="fichier" sourceLocation="#userprofil#\ndfaaaa_execAction\essai.xml"
targetLocation="#temp#\essai.xml" blockTagError="1"/>


</DEMO>
```

Créé avec HelpNDoc Personal Edition: Produire des livres Kindle gratuitement

Construction de module complémentaires

Construction de modules complémentaires

Ce chapitre détail la création de module complémentaires qui pourront être chargé ou déchargé à la volé dans le fichier de directive. Le module complémentaire à l'intérêt, une fois chargé, d'être manipulé comme une balise de directive, à avec quelques règles.

Règles de compatibilité :

- Les librairies qui devront être réalisées devront être en mode Dynamique (LIB ou DLL).
- Une fonction d'appel devra être présente (InitPlugIn) avec une structure de données.

Créé avec HelpNDoc Personal Edition: Créer des sites web d'aide facilement

Historique des versions

Historique des versions

Version actuelle :

4.10 :

- SESSION
  - Ajout du type actionquit
- REGISTRY
  - Amélioration de la suppression des clefs dans la base de registre
- Constantes
  - Ajout de l'abstraction timestamp en constante

- MEMO
  - Suppression du retour de la balise (-1)
- ACTION
  - Correction de l'attribut BlockAction
- LOG
  - Amélioration de l'affichage
  - Correction de l'archivage des fichiers

Historique :

0.1 :

Anciennement l'application se nommais IPR

0.572 :

- ACTION
- CHECK
- COPY
- DELETE
- EXEC
- REGISTRY
- ACTIVEDIRECTORY
- RENAMEDISK
- CREATE
- SHORTCUT
- CATCH
- DISKFORMAT
- SESSION
- MSGBOX

0.672 :

- CHECK
  - Ajout de fonctionnalités (vérification de processus en mémoire)
- REGISTRY
  - Mise en place de l'abstraction des chaînes de caractère.
- Amélioration de l'interface visuelle en ayant la possibilité de mettre des fond d'écran.
- MSGBOX
- SESSION
- DISKFORMAT

1.674 :

- COPY
  - Correction de bugs.
- EXEC
  - Correction de bugs.
- CATCH
  - Correction de bugs.
- Racine ACTION
  - Modification de l'affichage de l'image logo dans l'interface. Il est possible de mettre des chemins d'abstractions.

1.689 :

- REGISTRY
  - Correction de bugs sur le mode "add"

1.700 :

- SHORTCUT

1.825 :

- Racine ACTION
  - Ajout de fonctionnalité dans les paramètres.
  - Correction de bugs.
- COPY
  - Correction de bugs.
- Ajout du niveau d'affichage sur l'exécution du fichier de directives.
- CHECK
  - Correction de l'affichage.
- NETMAP

2.0 :

- REGISTRY
  - Ajout de la fonctionnalité de version par date.
  - Cette fonctionnalité entraîne une modification de l'emplacement du stockage en base de registre des informations de versions. Se referer à la rubrique de **racine action**.
- CHECK
  - Correction de bugs.
  - Ajout de la vérification par date de fichier.
- MAIL

- PAUSE

2.86 :

- ACTION
  - Correction de bugs, action en erreur alors que ignoré, défaut graphique.
- Racine ACTION
  - Ajout de la disposition de l'affichage wndowX et windowY.
- EXEC
  - Suppression de l'affichage du retour des applications.
  - Ajout de l'attribut *output*.

2.96 :

- Racine ACTION
  - Correction du redimensionnement de la fenêtre en mode "windows".
- EXEC
  - Correction de bugs.

2.111 :

- REGISTRY
  - Correction de l'édition de la base de registre
- CHECK
  - Correction de bugs.

2.444 :

- Correction de l'indentation des journaux d'exécution de fichier de directives.
- Correction de l'enchevêtrement des actions.

- MEMO
- MSGBOX
  - Modification de la mémorisation des retours.
- FILEBOX
- PATHBOX
- INPUTBOX

- DELETE
  - Modification de la vérification de la suppression des répertoires.

- REGISTRY
  - Ajout de l'accès au win64

- EXEC

- Correction du retour du programme.
- Ajout des fonctionnalités de mémorisation des codes retours.

ACTION

- Ajout des fonctionnalités de mémorisation des codes retour d'actions.

ACTIVEDIRECTORY

- Transformation du nom de la balise en LOCALUSERGROUP.

2.472 :

REGISTRY

- Amélioration sur la méthode de création de clefs et de valeurs.
- Remplacement de l'attribut "key" par "valuename".

CHECK

- Amélioration de la vérification sur la base de registre avec l'option *Mode*.
- Modification des incrémentations des niveaux.

2.558 :

CHECK

- Correction sur le parasitage de la balise dans une action.

Correction de l'écriture dans les fichiers journaux.
Correction de la documentation

2.566 :

REGISTRY

- Correction de la fermeture de clef dans la base de registre.

2.620 :

MEMO

- Ajout du type ini, chargement de fichier qui sera automatiquement mémorisé en variables.

- Amélioration du comportement des balise, sur la gestion des erreurs.

2.780 :

ACTION
- Correction de l'affichage

RENAMEDISK

Amélioration du comportement des balises, sur la gestion des erreurs.

3.0 :

- Version réservé.

4.0 :

RACINE ACTION
- Changement de l'interface graphique.
- Ajout de fonctionnalités pour le réglage de la fenêtre.
- Possibilité d'utiliser un fichier pour enregistrer les actions.

COMOGUI
WINDOW
FOR
WHILE
EXECACTION
WAIT
PAUSE
ACTION

Moteur ExecAction
- Modification du comportement des parcours des variables internes.

4.1 :

COMOGUI
LOG

4.2 :

MEMO
- Modification du comportement du chargement des fichiers INI

PLUGIN
LIB

4.8 :

- CATCH
  - balise renommée SCREENSHOT
- CHECK
  - Possibilité de rajouter plusieurs groupes avec l'option adgroup
- LIB
  - Correction de la gestion du code retour
- WAIT
  - Amélioration du comportement et correction du timeout
- DISKFORMAT

Créé avec HelpNDoc Personal Edition: Produire des livres EPub gratuitement

Légende de notation

Légende de notation

La description est composé de la version corrigé de l'application et des modifications effectué dans cette version.

- Version
  - Ajout d'une fonction
  - Modification d'une fonction
  - Suppression d'une fonction

Créé avec HelpNDoc Personal Edition: Générateur complet d'aides multi-formats