

# **Borland<sup>®</sup> StarTeam<sup>®</sup> 2009**

**Aide Ligne de commande de StarTeam**

**Borland<sup>®</sup>**

Borland Software Corporation  
8310 N Capital of Texas Hwy, Bldg 2, Ste 100  
Austin, Texas 78731 USA  
[www.borland.com](http://www.borland.com)

Les applications mentionnées dans ce document sont brevetées par Borland Software Corporation et/ou en attente de brevet. Reportez-vous au CD ou à la boîte de dialogue A propos pour avoir la liste des brevets. Le présent document ne donne aucun droit sur ces brevets.

Copyright © 1995–2009 Borland Software Corporation et/ou ses filiales. Tous les noms de marques et de produits Borland sont des marques de fabrique ou des marques déposées de Borland Software Corporation aux Etats-Unis et dans d'autres pays. Toutes les autres marques appartiennent à leurs propriétaires respectifs.

Juin 2009  
PDF

## Outils en ligne de commande

Référence .....	7
Opérations en ligne de commande .....	8
Options de ligne de commande de l'utilitaire Bulk Check-out .....	9
Paramètres de la commande starteamserver .....	17
Opérations en ligne de commande de l'utilitaire Checkout Trace .....	25
Options de ligne de commande de Vault Verify .....	26
Opérations en ligne de commande du client .....	28
Options communes .....	30
Caractères spéciaux .....	39
Codes de sortie .....	40
Ajouter des fichiers : stcmd add .....	41
Ajouter des dossiers : stcmd add-folder .....	43
Ajouter des projets : stcmd add-project .....	45
Ajouter des vues : stcmd add-view .....	47
Appliquer des étiquettes : stcmd apply-label .....	49
Archiver des fichiers : stcmd ci .....	50
Extraire des fichiers : stcmd co .....	53
Comparer les révisions d'un fichier : stcmd diff .....	56
Modifier les descriptions de fichiers : stcmd dsc .....	59
Créer des étiquettes : stcmd label .....	60
Créer des dossiers de travail : stcmd local-mkdir .....	62
Supprimer des fichiers locaux : stcmd delete-local .....	63
Afficher l'historique des fichiers : stcmd hist .....	64
Lister des fichiers : stcmd list .....	65
Verrouiller et déverrouiller des fichiers : stcmd lck .....	67
Verrouiller et déverrouiller un serveur : stcmd server-mode .....	68
Supprimer des fichiers : stcmd remove .....	70
Définir les options personnelles : stcmd set-personal-options .....	71
Mettre à jour l'état d'un fichier : stcmd update-status .....	73
Utilitaires VCM en ligne de commande .....	75
Présentation de l'utilitaire VCM en ligne de commande (VCMUtility) .....	76
Commandes de VCMUtility .....	80
Options de connexion de VCMUtility .....	83
Options de session de VCMUtility .....	85
Options diverses de VCMUtility .....	97
Exemples VCMUtility .....	98
Aide-mémoire .....	100
Syntaxe des options composées de VCMUtility .....	104
<action> .....	105
<options extraction> .....	106
<demandes de modification> .....	108
<fichiers> .....	109
<dossiers> .....	111
<type élément> .....	113
<état recherché> .....	114
<élément de processus> .....	116
<exigences> .....	117
<étiquettes de révision> .....	118
<tâches> .....	119
<horodatage> .....	120
<rubriques> .....	121





## Outils en ligne de commande

Cette section traite des divers outils permettant d'effectuer des opérations en ligne de commande.

### Dans cette section

[Référence](#)

Cette section contient toutes les rubriques de référence.

# Référence

Cette section contient des informations de référence.

## Dans cette section

### [Opérations en ligne de commande](#)

Cette section contient les rubriques de référence sur les opérations en ligne de commande.

### [Opérations en ligne de commande du client](#)

Cette section contient les rubriques de référence sur les opérations en ligne de commande du client.

### [Utilitaires VCM en ligne de commande](#)

Explique l'utilitaire Comparaison/Fusion de vues en ligne de commande (`VCMUtility`).

# Opérations en ligne de commande

Cette section contient les rubriques de référence sur les opérations en ligne de commande.

## Dans cette section

### [Options de ligne de commande de l'utilitaire Bulk Check-out](#)

Cette rubrique décrit les options de ligne de commande de l'utilitaire Bulk Check-out.

### [Paramètres de la commande starteamserver](#)

Décrit les options utilisées par la commande starteamserver et fournit des exemples.

### [Opérations en ligne de commande de l'utilitaire Checkout Trace](#)

Cette rubrique décrit les options de ligne de commande de l'utilitaire Checkout Trace.

### [Options de ligne de commande de Vault Verify](#)

Cette rubrique décrit les options de ligne de commande de l'utilitaire Vault Verify.



# Options de ligne de commande de l'utilitaire Bulk Check-out

Cette rubrique contient les informations suivantes concernant BCO :

- ◆ Syntaxe
- ◆ Options de ligne de commande
- ◆ Exemples d'utilisation de BCO

## Syntaxe

BCO utilise la syntaxe suivante.

```
-p "spécificateurProjet" [-pwdfile "cheminFichier"] [-autoLogon] [-cwf] [-is] [-rp "cheminDossier" | -fp "cheminDossier" ] [-cmp] [-dryrun] [-vb] [-useCA] [-encrypt typeChiffrement] [-cfgl "nomEtiquette"| -cfgp "nomEtat" | -cfgd "depuisDate"] [-filter "étatFichier"] [-o] [-ro] [-ts] [-fs] [-eol on | off | cr | lf | crlf ] [-exclude <modèle> | @<fichier modèle>] [-netmon] [-t] [-h | -help] [fichiers...]
```

## Options

Le tableau ci-dessous décrit les options utilisées avec BCO.

Option	Description
-autoLogon	Si aucun nom d'utilisateur n'est spécifié avec l'option <code>-p</code> , une tentative de connexion s'effectue avec l'ID et le mot de passe de l'utilisateur stockés par la barre d'outils pour le serveur spécifié. Cette fonctionnalité n'est disponible que sur les systèmes d'exploitation Windows.
-cfgd	Configure la vue à partir de la date et de l'heure spécifiées. Voici des exemples : <ul style="list-style-type: none"><li>■ "29/12/01 10:52 AM"</li><li>■ "December 29, 2001 10:52:00 AM PST"</li><li>■ "Monday, December 29, 2001 10:52:00 AM PST"</li></ul>
-cfgl	Configure la vue en utilisant l'étiquette spécifiée. Sans <code>-cfgl</code> , ni <code>-cfgp</code> , ni <code>-cfgd</code> , la configuration actuelle de la vue est utilisée.
-cfgp	Configure la vue en utilisant l'état de promotion spécifié.
-cmp	Comprime toutes les données envoyées de la station de travail au serveur et les décomprime une fois celles-ci arrivées. Sans cette option, aucune compression n'a lieu.  La compression est essentiellement utile et appropriée lorsque le client et le serveur communiquent via une connexion lente. Pour savoir s'il convient d'utiliser la compression, vous devez évaluer le cas de test suivant : le temps supplémentaire nécessaire à la compression et à la décompression des données est-il, ou non, inférieur au temps supplémentaire qui résulterait du transfert de données non compressées via la connexion lente ?
-cwf	Crée le dossier de travail pour le dossier de base et, si l'option <code>-is</code> est spécifiée, tous les sous-dossiers du dossier de base, même lorsqu'ils n'auront aucun fichier à extraire pendant l'exécution. Seuls les dossiers visibles sont créés.
-csf	Lorsque la commande associe le dossier spécifié par l'option <code>-p</code> au dossier StarTeam sous-jacent, l'utilisation de l'option <code>-csf</code> permet à la commande de différencier les dossiers StarTeam grâce à l'utilisation des minuscules/majuscules

dans leur nom (cette option ne modifie pas la distinction minuscules/majuscules pour les noms des fichiers appartenant à ces dossiers). Par exemple, avec `-csf`, les dossiers StarTeam nommés doc et Doc sont identifiés comme étant des dossiers différents. Sans cette option, les deux dossiers sont le dossier "doc".

Par défaut, les dossiers StarTeam ne sont pas différenciés d'après la casse des lettres de leur nom.

Avec ou sans l'option `-csf`, si des noms de dossiers sont ambigus, une erreur se produit. Par exemple, lorsque vous utilisez `-csf`, il y aura ambiguïté s'il existe à la fois un dossier Doc et un dossier doc. Lorsque vous n'utilisez pas `-csf`, il y a ambiguïté si les noms des dossiers s'écrivent de manière identique.

---

`-dryrun`

N'extrait pas les fichiers mais affiche la liste des fichiers qui seraient extraits si l'option `-dryrun` n'était pas spécifiée. Les chemins sont ceux des dossiers de travail dans lesquels les fichiers seraient copiés. Avec `-vb`, vous obtenez une image complète de ce qui se serait produit.

---

`-encrypt RC4 | RC2_ECB | RC2_CBC | RC2_CFB`

Chiffre toutes les données envoyées de la station de travail au serveur et les déchiffre une fois celles-ci arrivées. Sans cette option, aucun chiffrement n'a lieu. Le chiffrement empêche la lecture des fichiers et des autres informations de projet par des tiers non autorisés sur des lignes réseau non sécurisées.

La syntaxe complète est : `-encrypt typeChiffrement`

BCO prend en charge les types de chiffrement suivants :

*RC4* : Chiffrement en continu RSA RC4 (rapide)

*RC2\_ECB* : Chiffrement par bloc RSA RC2 (ECB)

*RC2\_CBC* : Chiffrement par bloc RSA RC2 (CBC)

*RC2\_CFB* : (Plates-formes Windows uniquement) Chiffrement par bloc RSA RC2 (CFB)

Ces types de chiffrement sont classés du plus rapide au plus lent. De deux types de chiffrement contigus, le plus lent est le plus sûr.

---

`-eol { on | off | cr | lf | crlf }`

Peut convertir automatiquement les marques de fin de ligne.

Lorsque spécifiée avec l'option `-on`, les fichiers texte sont transférés depuis le référentiel du serveur vers le dossier de travail de votre station de travail en respectant la convention de fin de ligne instaurée sur la plate-forme qui exécute la commande, et qui est déterminée par la VM Java.

Lorsque vous ne spécifiez pas cette option, ce qui est la valeur par défaut, BCO ne convertit pas les marques de fin de ligne. Utiliser la valeur par défaut équivaut à ne pas utiliser `-eol` du tout.

Lorsque vous spécifiez le caractère de fin de ligne (`cr`, `lf` ou `crlf`), BCO transfère les fichiers texte depuis le référentiel du serveur vers le dossier de travail de votre station de travail en respectant la convention de fin de ligne que vous avez spécifiée.

Pour les plates-formes Windows, la marque de fin de ligne est une combinaison retour chariot/saut de ligne (`crlf`) ; pour les plates-formes UNIX, c'est un saut de ligne (`lf`).

Par exemple, vous définirez cette option pour comparer un fichier du référentiel avec un fichier de travail présent sur un système UNIX (si le référentiel stocke les fichiers texte avec `crlf`).

---

`-exclude <modèle> | @<fichier modèle>`

Exclut les fichiers dont le nom ou le dossier parent correspond à un modèle donné. Un modèle peut être un nom exact de fichier ou de dossier ou contenir des caractères génériques (par exemple, `*.class`). Pour spécifier un nom de dossier, précédez le modèle d'une barre oblique (par exemple, `'/bin'`). Un seul modèle peut être fourni avec chaque `-exclude`, qui peut être répété. Sinon, un ou plusieurs

modèles peuvent être spécifiées sur des lignes distinctes du <fichier modèle> donné (préfixé par @).

---

fichiers...

Spécifie les fichiers à utiliser dans la commande, par leur nom ou par un modèle de nom de fichier (tel que "\*.c"). Toutes les options sont interprétées à l'aide des conventions sémantiques UNIX, et non de celles de Windows, car les conventions UNIX sont plus précises. De ce fait, "\*", plutôt que "\*.\*" signifie "tous les fichiers". Le modèle "\*.\*" signifie "tous les fichiers ayant une extension dans leur nom de fichier". Par exemple, le modèle "star\*.\*" trouve `starteam.doc` et `starteam.cpp` mais pas `starteam`. Pour trouver tous ces fichiers, il aurait fallu utiliser le modèle "star\*".

Sans cette option, la valeur par défaut est "\*".

Si, pour indiquer tous les fichiers, vous utilisez \* plutôt que "\*", un shell UNIX développe le modèle en une série d'éléments qu'il transmet à la commande bco sous forme d'un groupe d'options. Cela risquant de générer des problèmes (par exemple, lorsque vous extrayez des fichiers manquants), il est préférable d'utiliser "\*" pour les éviter.

Si vous utilisez plusieurs modèles de noms de fichiers, vous devez placer chacun d'eux dans son propre jeu de guillemets. Par exemple, vous pouvez utiliser "\*.bat" "\*.c", mais vous ne pouvez pas utiliser "\*.bat \*.c".

**Remarque :** Borland recommande de mettre cette option entre guillemets, quelle que soit la plate-forme, mais pour des raisons différentes. Sous les plates-formes Windows, les noms de fichiers et de dossiers qui contiennent des espaces ne sont pas correctement interprétés. Sous les plates-formes UNIX, le shell développe l'option, puis transmet au client la liste des éléments issus du développement. Cela génère fréquemment des résultats non voulus. Vous pouvez éviter ces deux conséquences en plaçant toujours cette option entre guillemets. Il n'est conseillé d'omettre les guillemets que s'il est essentiel que l'option soit développée par le shell UNIX. Les échecs inexpliqués des scripts de commandes par lot sur les deux plates-formes peuvent provenir à cette omission.

Plusieurs caractères spéciaux peuvent être utilisés dans la spécification des noms de fichiers :

\* Correspond à une chaîne, même vide. Par exemple, "\*" correspond à n'importe quel nom de fichier, avec ou sans extension. "xyz\*" correspond à "xyz", comme à "xyz.cpp" et à "xyzutyfj".

? Correspond à un caractère. Par exemple, "a?c" correspond à "abc", mais pas à "ac".

[...] Correspond à n'importe lequel des caractères qui sont placés entre les crochets.

Une paire de caractères séparés par un trait d'union (-) spécifie une plage de caractères à respecter.

Si le premier caractère qui suit un crochet ouvrant ( [ ) est un point d'exclamation ( ! ) ou un accent circonflexe ( ^ ), les caractères qui suivent ne sont pas recherchés. Tout caractère non placé entre les crochets est recherché. Par exemple, "x[a-d]y" correspond à "xby", mais pas à "xey". "x[!a-d]y" correspond à "xey", mais pas à "xby".

Un trait d'union (-) ou un crochet fermant ( ] ) peut être recherché si vous l'incluez en tant que premier ou dernier caractère d'un ensemble entre crochets.

Pour utiliser un astérisque (\*), un point d'interrogation (?) ou un crochet ouvrant ( [ ) dans un modèle, vous devez le faire précéder du caractère d'échappement (c'est-à-dire, d'une barre oblique inverse ( \ ).

---

-filter

Spécifie une chaîne d'un ou de plusieurs caractères, chacun d'entre eux représentant un état de fichier. N'incluez jamais d'espace ni aucune autre forme

d'espace vierge dans cette chaîne. Seuls les fichiers ayant les états spécifiés seront extraits. Vous ne pouvez pas extraire les fichiers dont l'état est *Pas dans la vue*.

Les lettres utilisées pour représenter les états sont les suivantes : **C** = (Current) A jour, **M** = Modifié, **G** = (merGe) Fusion, **O** = Obsolète, **I** = (mIssing) Manquant et **U** = (Unknown) Inconnu.

Si vous utilisez **C**, **M**, **G**, **O** ou **U**, vous devez aussi spécifier `-o` pour forcer l'opération d'extraction. Sans spécification de l'option `-o`, BCO omet les fichiers dont l'état est représenté par **C**, **M**, **G**, **O** ou **U**, mais un avertissement est consigné dans le fichier journal.

---

`-fp "cheminDossier"`

Redéfinit le dossier de travail, ou répertoire de travail, du dossier spécifié. Cela est équivalent à définir pour le dossier un chemin de travail secondaire.

Cette option, tout en vous permettant d'utiliser un dossier de travail différent de celui spécifié par le dossier, a pour principal objectif d'assurer la compatibilité multiplate-forme. Par exemple, les systèmes UNIX et Windows spécifient les noms de lecteurs et de chemins d'accès aux répertoires de manière incompatible. Le chemin `"D:\MYPRODUCT\DEVELOPMENT\SOURCE"` est compris par une plate-forme Windows mais ne l'est pas par une plate-forme UNIX. Utilisez cette option pour définir le chemin de travail si votre plate-forme ne comprend pas le chemin spécifié dans le projet.

Une barre oblique inverse ( `\` ) est interprétée comme caractère d'échappement lorsqu'elle précède des guillemets. Par conséquent, une erreur se produit dans l'exemple suivant :

```
bco -p "xxx" -fp "C:\" "*"
```

qui est interprété ainsi :

```
bco -p "xxx" -fp "C:" "*" .
```

Pour éviter ce type de situation, insérez un caractère d'échappement avant le dernier caractère de l'expression `"C:\"`, comme suit :

```
bco -p "xxx" -fp "C:\\\" "*" .
```

Vous pouvez également l'omettre, ainsi que dans l'exemple suivant, lorsque le chemin `-fp` ne finit pas par le dossier racine comme dans l'expression `"C:\orion\"` :

```
bco -p "xxx" -fp "C:\orion" "*" .
```

Remarquez également que les chemins de réseau UNC requièrent un caractère d'échappement supplémentaire ; par exemple :

```
bco -p "xxx" -fp "\\serveur\chemin".
```

La syntaxe complète est : `-fp "cheminDossier" .`

"Dossier" est le terme Windows et il apparaît dans l'interface utilisateur. "Répertoire" est le terme correct pour la plate-forme UNIX.

---

`-fs`

Empêche la mémorisation des états des fichiers après la procédure d'extraction. Par la suite, les valeurs d'état de ces fichiers seront incorrectes et imprécises. Utilisez cette option dans le cas où l'état d'un fichier n'est pas pertinent. Par exemple, si vous supprimez régulièrement les dossiers de travail avant d'extraire les fichiers pour créer un build, il n'y a pas de fichiers et donc leur état est sans importance.

Sachez que les états des fichiers peuvent ne jamais être connus — même si vous utilisez la commande `update-status` ultérieurement. Vous pouvez effectuer une extraction forcée sans l'option `-fs` pour que les fichiers en cours acquièrent l'état correct.

<code>-h</code>	Affiche des informations sur les options de la ligne de commande.
<code>-help</code>	
<code>-is</code>	Applique la commande de manière récursive à tous les dossiers enfant. Sans cette option, la commande s'applique uniquement au dossier spécifié.
<code>-netmon</code>	Dirige les informations du NetMonitor SDK vers la fenêtre console. NetMonitor affiche des statistiques pour les commandes de serveur. Voir les exemples pour BCO dans la section suivante de cette rubrique.
<code>-o</code>	Force l'extraction des fichiers dont l'état n'aurait, normalement, pas permis l'extraction. Ces états sont <i>Modifié</i> , <i>Fusion</i> ou <i>Inconnu</i> .
<code>-p</code>	<p>Indique la vue ou le dossier à utiliser et fournit le nom d'utilisateur et le mot de passe nécessaires pour accéder au serveur.</p> <p>La syntaxe complète est :</p> <pre><code>-p "nomUtilisateur:motPasse@nomHôte:numéroPort/ nomProjet/ [nomVue/] [hiérarchieDossiers/]"</code></pre> <p>Par exemple : <code>-p "JMarsh:motPasse@orion:49201/StarDraw/StarDraw/SourceCode/"</code>.</p> <p>Si le nom d'utilisateur est omis, le nom d'utilisateur actuel est utilisé. Le nom d'utilisateur spécifié dans l'exemple est <code>"JMarsh"</code>.</p> <p>Si le mot de passe est omis, l'utilisateur est invité à le saisir. Lorsque l'utilisateur saisit un mot de passe, les caractères ne sont pas affichés. Le mot de passe spécifié dans l'exemple est <code>"motPasse"</code>.</p> <p>Si le nom d'hôte est omis, la valeur par défaut est <code>localhost</code>. Le nom d'hôte spécifié dans l'exemple est <code>"orion"</code>.</p> <p>Le numéro de port est requis. L'exemple utilise le numéro de port par défaut : <code>49201</code>.</p> <p>Le nom de projet est toujours requis. Dans l'exemple, le nom de projet est <code>StarDraw</code>.</p> <p>Utilisez une hiérarchie de vues pour identifier la vue. Utilisez le signe "deux points" (<code>:</code>) comme délimiteur des noms de vues. La hiérarchie des vues doit toujours comprendre la vue racine. Par exemple, <code>"StarDraw:Release 4:Service Packs"</code> indique que la vue à utiliser est la vue <code>Service Packs</code>, qui est un enfant de la vue <code>Release 4</code> et un petit-enfant de la vue racine <code>StarDraw</code>. Si le nom de vue est omis, la vue racine (par défaut) est utilisée. Si seule cette vue porte ce nom dans ce projet, vous pouvez n'utiliser que le nom de la vue. (Cela n'est pas recommandé car une autre vue de même nom risque d'être créée ultérieurement et susciter des confusions.) Dans l'exemple, le nom de vue est <code>StarDraw</code>. Comme il s'agit de la vue racine du projet <code>StarDraw</code>, ce nom aurait pu être omis.</p> <p>Utilisez une hiérarchie de dossiers pour identifier le dossier. Utilisez la barre oblique (<code>/</code>) comme délimiteur des noms de dossiers. La hiérarchie de dossiers ne comprend jamais le dossier racine. Omettez la hiérarchie de dossiers si le fichier figure dans le dossier racine de la vue. Par exemple, si le dossier racine de la vue est <code>StarDraw</code> et que la hiérarchie d'accès à vos fichiers est <code>"StarDraw/SourceCode/Client"</code>, utilisez uniquement <code>"SourceCode/Client"</code>.</p> <p>Si l'un des paramètres utilisés avec cette option, par exemple, les noms des utilisateurs ou les noms des vues, contient les caractères servant de délimiteurs, vous devez utiliser le signe pourcentage (<code>%</code>) suivi du code hexadécimal correspondant, à la place de ces caractères. Par exemple, si <code>"@"</code> apparaît comme caractère dans un mot de passe, vous devez le remplacer par <code>"%40"</code>.</p> <p>Pour <code>":"</code>, utilisez <code>"%3a"</code>.</p>

Pour "/", utilisez "%2f".

Pour "@", utilisez "%40".

Pour "%", utilisez "%25".

Sous UNIX et d'autres systèmes d'exploitation, certains caractères spéciaux doivent être précédés d'une barre oblique inverse "\" ou d'un autre caractère d'échappement. Dans l'option `-p`, vous pouvez remplacer ces caractères par des codes hexadécimaux. Par exemple, "%3c" peut être utilisé sous UNIX à la place de "<".

Pour un espace, utilisez "%20".

Pour "<", utilisez "%3c".

Pour ">", utilisez "%3e".

---

`-pwdfile`

Spécifie le chemin d'accès du fichier qui stocke le mot de passe de l'utilisateur. Cette option redéfinit le mot de passe utilisé dans l'option `-p`. Elle empêche l'affichage du mot de passe de l'utilisateur dans la ligne de commande. Il doit être enregistré au format UTF-8.

La syntaxe complète est :

`-pwdfile "cheminFichier"`.

---

`-ro`

Rend le fichier de travail accessible uniquement en lecture après cette opération. Sans cette option, le fichier reste tel qu'il était avant l'opération. Habituellement, vous utiliserez `-ro` pour vous éviter de modifier un fichier non verrouillé par vous. Sans `-ro`, les fichiers sont accessibles en lecture et en écriture.

---

`-rp`

Spécifie ou redéfinit le dossier de travail, ou répertoire de travail, du dossier racine de la vue. Cela est équivalent à définir pour la vue un chemin de travail secondaire.

Pour de plus amples informations, reportez-vous dans le tableau à la description de l'option `-fp`.

La syntaxe complète est :

`-rp "cheminDossier"`.

---

`-t`

Affiche des statistiques sur les volumes et les durées d'extraction.

---

`-ts`

Définit l'horodatage de chaque fichier de travail par l'heure de l'extraction. Sans cette option, le fichier reçoit le même horodatage que sa révision archivée.

---

`-useCA`

Tente d'extraire les fichiers en utilisant un Agent de mise en cache StarTeamMPX.

La syntaxe complète est :

`-useCA host:port | autolocate`.

La syntaxe `hôte:port` spécifie le nom d'hôte (ou adresse IP) et le numéro de port de l'Agent de mise en cache à utiliser.

Alternativement, `autolocate` peut être spécifiée pour localiser automatiquement l'Agent de mise en cache le plus proche sur le réseau. `autolocate` nécessite que le serveur StarTeam soit activé pour MPX.

---

`-vb`

La sortie est explicite. Affiche le nom de chaque fichier lors de son extraction. Le chemin du dossier est le chemin du dossier et non celui du dossier de travail.

---

## Exemples d'utilisation de BCO

L'exemple suivant utilise BCO pour forcer l'extraction de tous les fichiers situés dans *Source Code*, enfant du dossier racine *StarDraw* (dans la vue *StarDraw* du projet *StarDraw*).

```
bco -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/Source Code" -is -o ""
```

L'exemple suivant montre une commande BCO utilisant l'option `-netmon`, ainsi que la sortie générée affichée par NetMonitor.

```
bco -p "Administrator:Administrator@10.50.6.91:49201/StarDraw/WebSite" -fp D:\ Test  
-netmon -o "*.htm"
```

### Exemple de sortie :

```
StarTeam BulkCheckOut Utility version 9.0.xxx  
Copyright (c) 2006 Borland Software Corporation. Tous droits réservés.  
Start: (rev 100) SRVR_CMD_GET_PROJECT_LIST Time: 62 millis; Sent: 42 bytes;  
Got: 1834 bytes  
Start: (rev 100) SRVR_CMD_GET_PROJECT_VIEWS Time: 47 millis; Sent: 46 bytes;  
Got: 186 bytes  
Start: (rev 100) SRVR_CMD_GET_PROJECT_VIEWS Time: 15 millis; Sent: 46 bytes;  
Got: 186 bytes  
Start: (rev 100) SRVR_CMD_PROJECT_OPEN Time: 188 millis; Sent: 70 bytes;  
Got: 120 bytes  
Start: (rev 100) PROJ_CMD_GET_VIEW_PROPERTIES Time: 31 millis; Sent: 42 bytes;  
Got: 2556 bytes  
Start: (rev 100) PROJ_CMD_GET_FOLDERS Time: 63 millis; Sent: 42 bytes;  
Got: 1112 bytes  
Start: (rev 100) PROJ_CMD_GET_FOLDER_ITEMS Time: 16 millis; Sent: 50 bytes;  
Got: 40 bytes  
Start: (rev 100) PROJ_CMD_REFRESH_ITEMS Time: 3562 millis; Sent: 122 bytes;  
Got: 414 bytes  
Start: (rev 100) SRVR_CMD_GET_PROJECT_VIEWS Time: 16 millis; Sent: 46 bytes;  
Got: 186 bytes  
Start: (rev 100) PROJ_CMD_GET_PROJECT_PROPERTIES Time: 31 millis; Sent: 42 bytes;  
Got: 4797 bytes  
Start: (rev 100) FILE_CMD_CHECKOUT Time: 47 millis; Sent: 78 bytes;  
Got: 108 bytes  
Start: (rev 100) FILE_CMD_CHECKOUT Time: 31 millis; Sent: 78 bytes;  
Got: 1767 bytes  
Start: (rev 100) FILE_CMD_CHECKOUT Time: 31 millis; Sent: 78 bytes;  
Got: 1140 bytes  
Start: (rev 100) SRVR_CMD_PROJECT_CLOSE Time: 15 millis; Sent: 62 bytes;
```

Got: 16 bytes

Start: (rev 100) SRVR\_CMD\_RELEASE\_CLIENT Time: 31 millis; Sent: 42 bytes;

Got: 16 bytes



## Paramètres de la commande `starteamserver`

Cette section décrit par ordre alphabétique les options de la commande `starteamserver`, en fournissant des exemples d'utilisation.

### **-access Clé**

Utiliser avec : `-serial`.

Voir aussi : `-serial`, `-license` et `-eval`.

Enregistre le serveur avec une version sous licence. Utilisez cette option avec l'option `-serial`. La première fois que vous démarrez le serveur, vous devez enregistrer l'application en tant que version sous licence ou copie d'évaluation. Pour obtenir une combinaison numéro de série/clé d'accès ou une clé d'évaluation afin d'allonger la période d'évaluation, contactez <http://www.borland.com/us/company/how-to-buy.html>.

Exemple :

```
starteamserver -serial 1234 -access 5678
```

### **-all**

Utiliser avec : `-start`, `-stop` et `-restart`.

Utilisée conjointement aux options `-start` (ou `-restart`) ou `-stop`. Les options `-start -all` démarrent toutes les configurations de serveur dont l'état est Prêt (Ready) dans le fichier `starteam-server-configs.xml`. Les options `-stop` et `-all` arrêtent toutes les configurations de serveur dont l'état est Exécution (Running).

Exemple :

```
starteamserver -stop -all
```

### **-attach "CheminPiècesJointes"**

Utiliser avec : `-start` et `-restart`.

Indique le chemin d'accès aux pièces jointes d'une configuration de serveur.

Lors de la première exécution d'une configuration de serveur, le système crée un dossier enfant Attachments dans le CheminRéférentiel (RepositoryPath) que vous indiquez, et il stocke le chemin d'accès à ce dossier dans la base de données utilisée par la configuration du serveur. Si vous avez modifié l'emplacement du dossier Attachments, vous pouvez modifier le chemin d'accès aux pièces jointes dans la base de données à l'aide des options de la ligne de commande `-start` (ou `-restart`) et `-attach`.

Il est également possible de modifier le chemin d'accès aux pièces jointes sur la page Général de la boîte de dialogue Configuration du serveur StarTeam. Le nouveau chemin des pièces jointes sera pris en compte lors du prochain démarrage de la configuration du serveur.

Exemple :

```
starteamserver -start MonServeur -attach "c:\MonServeur\Attachments"
```

## **-autorecover**

Utiliser avec : `-start`.

Voir aussi : `-stoponerrors`.

L'option `-autorecover` ordonne au serveur de tenter d'apporter des réparations, de nature limitée, pendant le processus de vérification quand cela s'avère nécessaire.

Exemple :

```
starteamserver -start MonServeur -autorecover
```

## **-dsn "NomSourceDonnées"**

Utiliser avec : `-new`, `-edit`, `-start` et `-restart`.

Voir aussi : `-t`, `-p`, et `-u`.

Indique les informations de connexion à la base de données. Entrez le nom de la source de données (DSN) ODBC existant.

Dans les versions 5.1 et 5.2, l'accès aux bases de données Oracle s'effectuait via le nom de service réseau d'Oracle stocké dans `$ORACLE_HOME/network/admin/tnsnames.ora`. Cela n'est plus le cas.

La valeur que vous spécifiez pour `NomServeurBD` (`DBServerName`) est stockée dans le fichier `starteam-server-configs.xml`. Pour afficher ou modifier les informations de connexion à la base de données, vous pouvez utiliser :

- ◆ Les options `-view` et `-edit` à partir de la ligne de commande.
- ◆ L'onglet Base de données de la boîte de dialogue **Configuration du serveur StarTeam** dans Administration du serveur.
- ◆ L'onglet Base de données de la boîte de dialogue Propriétés <configuration de serveur> dans Administration du serveur.

Les modifications prennent effet après redémarrage du serveur.

Exemple :

```
starteamserver -edit MonServeur -dsn DSNMonServeur
```

## **-edit "NomConfiguration"**

Utiliser avec : `-name`, `-dsn`, `-u` et `-p`.

Modifie les options de session de la configuration de serveur indiquée. Vous pouvez modifier les options suivantes : `-name`, `-dsn`, `-u`, `-p`. Si la configuration de serveur s'exécute, vous devez l'arrêter pour effectuer les modifications.

Exemple :

```
starteamserver -edit MonServeur -name Portable -dsn ServeurDistant  
-u StarTeamAdmin
```

## **-eval Numéro**

Voir aussi : `-serial`, `-access` et `-license`.

Allonge la période d'évaluation d'une copie d'évaluation du serveur. La première fois que vous démarrez le serveur, vous devez enregistrer l'application en tant que version sous licence ou copie d'évaluation. Pour obtenir une combinaison numéro de série/clé d'accès ou une clé d'évaluation afin d'allonger la période d'évaluation, contactez <http://www.borland.com/us/company/how-to-buy.html>.

Exemple :

```
starteamserver -eval 01234567890
```

## **-help**

Affiche un message décrivant toutes les options de la commande.

Exemple :

```
starteamserver -help
```

## **-licenses**

Voir aussi : `-serial`, `-access` et `-eval`.

Affiche les informations de licence et d'enregistrement. Si vous exécutez une copie d'évaluation de l'application, le système affiche un message pour vous en informer. Dans le cas contraire, le système affiche le numéro de série.

Exemple :

```
starteamserver -licenses
```

## **-list**

Dresse la liste des configurations de serveur définies dans le fichier `starteam-server-configs.xml`, accompagnées de leur état. A un instant donné, une configuration de serveur peut se trouver dans l'un des états suivants : Prêt, Démarrage, Exécution, Désactivé et Arrêt.

Exemple :

```
starteamserver -list
```

Le serveur affiche un message similaire à celui-ci :

```
Etat des configurations : MonServeur Prêt, StarDrawRepository Exécution  
Portable Prêt
```

## **-name "NomConfiguration"**

Utiliser avec : `-edit`, `-start` et `-restart`.

Renomme une configuration de serveur. Cette option est utilisée conjointement à l'option `-edit`. Le nouveau nom de la configuration de serveur prendra effet après le redémarrage de celle-ci.

Exemple :

```
starteamserver -edit MonServeur -name ServeurNouvelleEquipe
```

## **-new "NomConfiguration"**

Crée un répertoire de stockage nommé DefaultHive pour la nouvelle configuration de serveur, avec le nom et les paramètres spécifiés. Cette configuration utilise un espace de stockage Native-II. Cette option produit le même résultat que sélectionner **Nouveau** dans le menu de l'utilitaire Administration du serveur et utiliser l'expert permettant de créer une nouvelle configuration.

Certaines options ne sont utilisables qu'avec `-new`. Ce sont : `-c`, `-r` et `-t`.

Exemple :

```
starteamserver -new NouveauServeur1 -r "c:\nouveau serveur\" -t 1 -dsn DSNNouveauServeur  
-u Admin -p motPasse
```

## **-p "MotPasseUtilisateurBD"**

Utiliser avec : `-new`, `-edit`, `-start` et `-restart`.

Voir aussi : `-dsn`, `-p`, `-t` et `-u`.

Indique le mot de passe d'accès à la base de données. La valeur que vous spécifiez pour MotPasseUtilisateurBD (DBPassword) est stockée dans le fichier `starteam-server-configs.xml`. Assurez-vous que le mot de passe que vous indiquez est bien associé au nom de l'utilisateur de la base de données. Vous pouvez examiner ou modifier le mot de passe de l'utilisateur de la base de données à l'aide des options `-view` et `-edit` à partir de la ligne de commande. Les modifications que vous effectuez seront prises en compte lors du prochain démarrage de la configuration du serveur.

Exemple :

```
starteamserver -edit MonServeur -u JodyK -p motPasse
```

## **-q**

Exécute une commande en mode silencieux — ce qui signifie que le système n'affiche pas de sortie.

Exemple :

```
starteamserver -stop -all -q
```

## **-r "CheminRéférentiel"**

Utiliser avec : `-new`.

Indique le chemin d'accès au référentiel d'une configuration de serveur. Si le chemin d'accès du référentiel n'existe pas, le système crée les dossiers appropriés lors de la première exécution de cette configuration de serveur.

La valeur que vous spécifiez pour `CheminRéférentiel` (RepositoryPath) est stockée dans le fichier `starteam-serverconfigs.xml`. Pour afficher et modifier le chemin d'accès au référentiel, vous pouvez utiliser l'option `-view` de la ligne de commande ou, depuis l'application, l'onglet **Général** de l'utilitaire Configuration du serveur StarTeam dans l'utilitaire Administration du serveur.

**Avertissement :** Evitez d'utiliser le dossier/répertoire de base du serveur comme chemin du référentiel d'une configuration de serveur car celle-ci ne démarrerait pas.

Exemple :`starteamserver -new NouveauServeur1 -r "c:\nouveau serveur\" -t 1 -dsn DSNNouveauServeur -u Admin -p motPasse`

## **-remove "NomConfiguration"**

Supprime du fichier `starteam-server-configs.xml` la configuration de serveur indiquée.

Exemple :

```
starteamserver -remove MonServeur
```

## **-restart "NomConfiguration"**

Arrête et redémarre la configuration de serveur spécifiée. Utilisez cette option pour valider les modifications que vous avez apportées à une configuration de serveur. En cas d'échec du redémarrage de la configuration du serveur, consultez le fichier journal du serveur.

Vous pouvez relancer une configuration de serveur et modifier un certain nombre de ses options simultanément. Les options suivantes sont utilisables avec l'option `-restart` : `-all`, `-attach`, `-dsn`, `-name`, `-p`, `-tcpip` et `-u`. L'option `-all` et le nom d'une configuration ne sont pas utilisables simultanément.

Exemple :

```
starteamserver -restart MonServeur -tcpip TCPIPStarTeam -u SuperUtilisateur  
-p MotPasseSuperUtilisateur
```

## **-serial Numéro**

Voir aussi : `-access`, `-license` et `-eval`.

Enregistre le serveur avec une version sous licence. Utilisez cette option avec l'option `-access`. La première fois que vous démarrez le serveur, vous devez enregistrer l'application en tant que version sous licence ou copie d'évaluation. Pour obtenir une combinaison numéro de série/clé d'accès ou une clé d'évaluation afin d'allonger la période d'évaluation, contactez <http://www.borland.com/us/company/how-to-buy.html>. Les numéros de série et d'accès de cet exemple doivent être remplacés par des valeurs réelles.

Exemple :

```
starteamserver -serial 1234567890 -access 9999999
```

## **-start "NomConfiguration"**

Voir aussi : `-all` et `-stop`.

Lance la configuration de serveur indiquée. `starteamserver` met à jour l'entrée de la configuration de serveur dans le fichier `starteam-server-configs.xml` avec `Status=Running` et `PID=nnn` où `nnn` doit être remplacé par le numéro de `PID` réel.

Vous pouvez lancer une configuration de serveur et modifier un certain nombre de ses options simultanément. Les options suivantes sont utilisables avec l'option `-start` : `-attach`, `-dsn`, `-name`, `-p`, `-tcpip` et `-u`.

Exemple :

```
starteamserver -start MonServeur -tcpip TCPIPStarTeam -u SuperUtilisateur  
-p MotPasseSuperUtilisateur
```

## **-stop "NomConfiguration"**

Voir aussi : `-all` et `-start`.

Arrête la configuration de serveur indiquée. Après l'arrêt de la configuration de serveur, `starteamserver` met à jour l'entrée du fichier `starteam-server-configs.xml` avec `Status=Ready` et `PID=0`.

Exemple :

```
starteamserver -stop MonServeur
```

**Remarque :** Remarque à l'attention des utilisateurs d'Enterprise Advantage : Si vous exécutez le serveur comme service et l'Agent de notification comme service dépendant, vous ne pouvez pas arrêter le serveur à moins de fermer d'abord le service Agent de notification.

## **-t TypeBD**

Utiliser avec : `-new`

Voir aussi : `-dsn`, `-p`, `-u`.

Spécifie le type de la base de données. Cette option n'est utilisable que lorsque vous créez une configuration de serveur. Indiquez le type de la base de données à l'aide de l'une des valeurs numériques suivantes :

2 = Microsoft SQL Server ou SSE

3 = Oracle

La valeur que vous spécifiez pour `TypeBD` (`DBType`) est stockée dans le fichier `starteam-server-configs.xml`. Pour afficher le type de la base de données, vous disposez des moyens suivants :

- ◆ L'option `-view` à partir de la ligne de commande.
- ◆ L'onglet **Base de données** de l'utilitaire **Configuration du serveur StarTeam** dans l'utilitaire Administration du serveur.
- ◆ L'onglet **Base de données** de la boîte de dialogue **Propriétés** `<configuration de serveur>` dans Administration du serveur.

Exemple :

```
starteamserver -new NouveauServeur1 -r "c:\nouveau serveur\" -t 1 -dsn DSNNouveauServeur
-u Admin -p motPasse
```

## **-tcpip pointTerminaison | up[:pointTerminaison] | down[:pointTerminaison]**

Utiliser avec : `-start`, `-restart`

Définit le point de terminaison du protocole TCP/IP (Sockets). Active ou désactive également le protocole. Utilisez `up` pour l'activation et `down` pour la désactivation. Pour définir le point de terminaison et l'activer ou le désactiver, indiquez l'option `up` ou l'option `down` suivie du signe deux-points et du point de terminaison.

La valeur indiquée pour le point de terminaison (Endpoint) est stockée dans la base de données employée par cette configuration de serveur.

Vous pouvez modifier cette information à l'aide des options `-start` (ou `-restart`) et `-tcpip` sur la ligne de commande ou, depuis l'application, sur l'onglet **Protocole** de la page **Configuration du serveur StarTeam**.

Exemple :

```
starteamserver -start MonServeur -tcpip 49201 starteamserver -start
MonServeur -tcpip up
```

## **-u "NomUtilisateurBD"**

Utiliser avec : `-new`, `-edit`, `-start`, `-P` et `-restart`.

Voir aussi : `-t`, `-dsn` et `-p`.

Indique le nom d'utilisateur que la configuration de serveur emploie pour accéder à la base de données. La valeur que vous spécifiez pour `NomUtilisateurBD` (DBUserName) est stockée dans le fichier `starteam-server-configs.xml`. Vous pouvez examiner ou modifier le nom de l'utilisateur de la base de données à l'aide des options `-view` ou `-edit` de la ligne de commande. Assurez-vous d'avoir indiqué le mot de passe de ce compte utilisateur. Les modifications que vous effectuez seront prises en compte lors du prochain démarrage de la configuration du serveur. Assurez-vous que le nom d'utilisateur et le mot de passe indiqués dans la commande `starteamserver` correspondent à un compte de la base de données. La configuration de l'utilisateur ne pourra pas démarrer si le compte utilisateur ne figure pas dans la base de données.

Exemple :

```
starteamserver -edit MonServeur -u SuperUtilisateur -p motPasseSuperUtilisateur
```

## **-version**

Affiche le numéro de version et le numéro de build du serveur.

Exemple :

```
starteamserver -version
```

Le serveur affiche un message similaire à celui-ci :

```
Version StarTeam Server : x.x Numéro de build : x.x.xxx
```

### **-view "NomConfiguration"**

Dresse la liste des propriétés de session de la configuration de serveur indiquée.

Exemple :

```
starteamserver -view StarDraw
```

### **Référence associée**

[Opérations en ligne de commande](#)



# Opérations en ligne de commande de l'utilitaire Checkout Trace

Cette rubrique décrit les options de ligne de commande de l'utilitaire Checkout Trace.

En général, vous pouvez exécuter l'utilitaire depuis la ligne de commande avec les options par défaut, comme suit : `CheckoutTraceDump.exe -go`. Les options valides de Check-out Trace sont décrites dans le tableau suivant.

Option	Description
<code>-go</code>	Spécifiez cet indicateur pour exécuter les valeurs par défaut.
<code>-path:&lt;chemin&gt;</code>	Le dossier contenant les fichiers binaires de suivi des extractions ( <code>.cotrc</code> ). La valeur par défaut est le dossier en cours.
<code>-outpath:&lt;chemin&gt;</code>	Le dossier qui contiendra les fichiers générés en sortie ( <code>.csv</code> ). La valeur par défaut est le dossier contenant les fichiers binaires de suivi ( <code>.cotrc</code> ).
<code>-file:&lt;spécFic&gt;</code>	Les fichiers binaires de suivi des extractions à utiliser en entrée. Prend en charge les caractères génériques standard des systèmes de fichiers (*, ?). La valeur par défaut est <code>"*.cotrc"</code> (tous les fichiers de suivi des extractions contenus dans le dossier). Vous ne pouvez pas utiliser plus d'un chemin avec ce paramètre et vous ne pouvez pas spécifier ce paramètre plus d'une fois par commande.
<code>-ext:&lt;extension&gt;</code>	L'extension de fichier utilisée pour les fichiers de suivi des extractions. L'extension est ajoutée au nom de fichier binaire de suivi des extractions ( <code>.cotrc</code> ) afin de créer le nom du fichier de vidage. La valeur par défaut est <code>.csv</code> .
<code>-start:&lt;date début&gt;</code>	La date-heure la plus ancienne à sélectionner. Seules les extractions s'étant produites après cette date seront dans la sortie. Par défaut, l'utilitaire n'effectue pas de filtrage sur les dates.
<code>-end:&lt;date fin&gt;</code>	Spécifie la date-heure la plus récente à sélectionner. Seules les extractions s'étant produites avant cette date seront dans la sortie. Par défaut, l'utilitaire n'effectue pas de filtrage sur les dates.
<code>-project:&lt;nom du projet&gt;</code>	Nom du projet dans lequel se trouvent les informations d'extraction à filtrer. Seules les extractions de ce projet seront dans la sortie. Par défaut, l'utilitaire n'effectue pas de filtrage sur les projets. Tous les projets sont inclus dans la sortie. Lorsque, à la fois l'option <code>-project</code> et l'option <code>-projectid</code> sont spécifiées, <code>-projectid</code> l'emporte.
<code>-projectid:&lt;ID du projet&gt;</code>	ID du projet dans lequel se trouvent les informations d'extraction à filtrer. Seules les extractions de ce projet seront dans la sortie. Par défaut, l'utilitaire n'effectue pas de filtrage sur l'ID des projets. Tous les projets sont inclus dans la sortie. Cette option l'emporte sur <code>-project</code> lorsque les deux options sont spécifiées.
<code>-separator:&lt;séparateur&gt;</code>	Chaîne servant à séparer les valeurs dans le fichier de sortie. Par défaut, l'utilitaire utilise <code>" , "</code> .
<code>-overwrite</code>	Spécifiez cet indicateur pour écraser les fichiers de suivi des extractions existants. S'il n'est pas spécifié, les fichiers binaires de suivi des extractions seront omis quand il existe déjà un fichier de vidage de suivi portant le nom concerné.

# Options de ligne de commande de Vault Verify

Cette rubrique décrit les options de ligne de commande de l'utilitaire Vault Verify.

En général, vous pouvez exécuter Vault Verify depuis la ligne de commande, comme suit : `VaultVerify [options] "configuration"`.

Selon l'option `-check` utilisée, qu'elle soit par défaut ou définie, les vérifications d'intégrité sont effectuées sur les fichiers d'archive de l'espace de stockage correspondant à la "configuration de serveur" StarTeam spécifiée. Si vous spécifiez l'option `-repair`, Vault Verify tente de corriger les problèmes trouvés. Vault Verify ouvre la base de données correspondant à la configuration de serveur via JDBC, mais il ne la modifie pas. Les options valides de Vault Verify sont décrites dans le tableau suivant.

Option	Description
<code>-check {missing   corrupt   stray   all}</code>	Détermine quelles vérifications d'intégrité effectuer :  <code>missing</code> : Vérifie les fichiers manquants en comparant la base de données et les fichiers d'archive réellement présents.  <code>corrupt</code> : Vérifie l'intégrité des fichiers d'archive existants (MD5, nom, dossier et format de fichier .gz).  <code>stray</code> : Vérifie les fichiers en supplément par rapport à la base de données. Cette option ne peut pas être utilisée si la configuration de serveur est en cours d'utilisation.  <code>all</code> : Effectue toutes les vérifications d'intégrité.  Plusieurs options <code>-check</code> peuvent être spécifiées. Voir aussi l'option <code>-repair</code> .
<code>-cf &lt;chemin du dossier&gt;</code>	Nom du chemin d'accès au <i>dossier des fichiers endommagés</i> , où sont déplacés les fichiers problématiques trouvés par la vérification <code>corrupt</code> lorsque l'option <code>-repair</code> a été spécifiée. Par défaut, le dossier des fichiers endommagés est le dossier <code>C:\Temp\VVCorruptFiles</code> .
<code>-dbhost &lt;hôte&gt;</code>	Spécifie le nom d'hôte de la base de données pour la <configuration de serveur> spécifiée. Sous Windows, cela n'a de sens que si <code>-dbinstance</code> est également fourni. Sous Windows et Linux, utilisez cette option uniquement lorsque le serveur de base de données s'exécute sur un hôte différent.
<code>-dbname &lt;nom&gt;</code>	Spécifie le nom de la base de données pour la <configuration> indiquée. Sous Windows, ce paramètre n'est significatif que si <code>-dbinstance</code> est également spécifié et il n'est nécessaire que si le nom de la base de données est différent du DSN ODBC. Sous Linux, utilisez cette option uniquement si <code>-dbinstance</code> n'est pas utilisé et si le nom de service Oracle est différent du nom TNS.
<code>-dbinstance &lt;nom&gt;</code>	Cette option n'a de sens que sous Windows. Lorsqu'elle est utilisée, VaultVerify ouvre la base de données directement et non via le DSN ODBC spécifié dans le fichier de configuration. Pour SQL Server, le <nom> doit être le nom d'instance (par exemple, 'SSE2005_ST'). Notez que le nom d'instance par défaut est 'MSSQLSERVER' pour SQL Server et 'SQLEXPRESS' pour SQL Server Express. Pour Oracle, ce doit être le nom de service (par exemple, 'ORCL').  <code>-dbinstance</code> doit être utilisé avec <code>-dbhost</code> lorsque le serveur de la base de données s'exécute sur un hôte différent. Pour SQL Server, <code>-dbname</code> doit aussi être utilisé si le nom de la base de données est différent du DSN ODBC. Pour Oracle, <code>-dbname</code> est ignoré si <code>-dbinstance</code> est spécifié.

<code>-dbpassword &lt;motPasse&gt;</code>	Spécifie le mot de passe de connexion à la base de données. S'il n'est pas spécifié, un mot de passe vierge est utilisé. (Le mot de passe stocké dans la configuration est chiffré et il ne peut pas être utilisé par Vault Verify.) Sur les configurations de serveur s'exécutant avec Oracle, cette option doit être spécifiée car le mot de passe Oracle n'est jamais vide.
<code>-dbport &lt;port&gt;</code>	Spécifie le port TCP/IP à utiliser pour se connecter au serveur de la base de données. Ce paramètre ne sert qu'aux plates-formes non Windows lorsqu'un autre port que le port de la base de données par défaut du fournisseur est utilisé (par exemple, 1521 pour Oracle).
<code>-dbuser &lt;utilisateur&gt;</code>	Spécifie l'ID de connexion à utiliser pour se connecter à la base de données. S'il est spécifié, ce paramètre remplace l'utilisateur indiqué dans la <configuration> StarTeam. Le seul utilisateur valide pour cette option est l'utilisateur propriétaire des tables StarTeam.
<code>-help (ou -h ou -?)</code>	Affiche ces informations d'utilisation.
<code>-path &lt;chemin du dossier&gt;</code>	Spécifie le chemin du dossier contenant le fichier <code>starteam-server-configs.xml</code> . Ce fichier doit exister et contenir la <configuration de serveur> spécifiée. Par défaut, ce fichier est ouvert dans le dossier parent du répertoire de travail en cours s'il n'est pas trouvé dans le répertoire de travail en cours.
<code>-nosharereport</code>	Supprime le reporting des informations de partage. Normalement, les chemins partagés de chaque fichier endommagé sont consignés dans un rapport. Cette option supprime les informations concernant les chemins partagés ; cela peut accélérer l'exécution de l'application et réduire de manière significative la taille des rapports.
<code>-repair</code>	Spécifie que la correction des problèmes des fichiers d'archive doit être tentée. Les archives endommagées sont déplacées dans le 'dossier des fichiers endommagés' (voir l'option <code>-cf</code> ). Si elles correspondent à des révisions de fichiers valides, elles sont également traitées comme étant des fichiers manquants. L'utilitaire tente de récupérer les archives manquantes depuis d'autres fichiers de l'espace de stockage et, si l'option <code>-useca</code> est spécifiée, depuis l'Agent de mise en cache. Les archives isolées sont déplacées dans le 'dossier des fichiers isolés' (voir l'option <code>-sf</code> ).  <b>Remarque :</b> L'option <code>-repair</code> est ignorée si la <configuration de serveur> StarTeam est en cours d'utilisation.
<code>-sf &lt;chemin du dossier&gt;</code>	Nom du chemin d'accès au 'dossier des fichiers isolés', où sont déplacés les fichiers supplémentaires trouvés par la vérification 'stray' lorsque l'option <code>-repair</code> a été spécifiée. Par défaut, le 'dossier des fichiers isolés' est le dossier <code>C:\Temp\VVStrayFiles</code> .
<code>-t</code>	Affiche des informations sur le temps écoulé à la fin de la vérification.
<code>-useca &lt;hôte&gt;:&lt;port&gt;</code>	Si <code>-repair</code> est spécifié, cette option permet de tenter de récupérer les fichiers manquants depuis l'Agent de mise en cache MPX spécifié. L'<hôte> et le <port> doivent désigner un Agent de mise en cache distant car celui-ci maintient un cache indépendant.
<code>-verbose</code>	Affiche des informations d'état supplémentaires pendant le déroulement de la vérification.
<code>"configuration"</code>	Spécifie le nom de la configuration. Le nom de la configuration transmis à Vault Verify distingue les majuscules des minuscules et, s'il contient des espaces, doit être placé entre guillemets.

# Opérations en ligne de commande du client

Cette section contient les rubriques de référence sur les opérations en ligne de commande du client.

## Dans cette section

### [Options communes](#)

Décrit les options apparaissant dans toutes les commandes, ou presque.

### [Caractères spéciaux](#)

Décrit les caractères spéciaux pouvant servir à la recherche des fichiers.

### [Codes de sortie](#)

Décrit les codes de sortie renvoyés par les commandes stcmd :

### [Ajouter des fichiers : stcmd add](#)

Décrit la commande permettant d'ajouter des fichiers à un projet.

### [Ajouter des dossiers : stcmd add-folder](#)

Décrit la commande permettant d'ajouter des dossiers à une vue.

### [Ajouter des projets : stcmd add-project](#)

Décrit la commande permettant d'ajouter un projet à une configuration de serveur StarTeam.

### [Ajouter des vues : stcmd add-view](#)

Décrit la commande permettant d'ajouter une vue à une configuration de serveur StarTeam.

### [Appliquer des étiquettes : stcmd apply-label](#)

Décrit la commande servant à étiqueter les révisions de fichiers spécifiées en utilisant des étiquettes de vue ou de révision.

### [Archiver des fichiers : stcmd ci](#)

Décrit la commande permettant d'archiver des fichiers dans un référentiel (ou espace de stockage) StarTeam à partir d'un dossier de travail.

### [Extraire des fichiers : stcmd co](#)

Décrit la commande permettant d'extraire des fichiers à partir d'un référentiel (ou espace de stockage) StarTeam dans un dossier de travail.

### [Comparer les révisions d'un fichier : stcmd diff](#)

Décrit la commande permettant d'afficher les différences entre deux révisions d'un fichier.

### [Modifier les descriptions de fichiers : stcmd dsc](#)

Décrit la commande permettant de modifier la description d'un fichier.

### [Créer des étiquettes : stcmd label](#)

Décrit la commande permettant de créer une étiquette de vue ou de révision.

### [Créer des dossiers de travail : stcmd local-mkdir](#)

Décrit la commande permettant de créer sur votre station de travail le dossier de travail ou le répertoire de travail pour le dossier StarTeam spécifié.

### [Supprimer des fichiers locaux : stcmd delete-local](#)

Décrit la commande permettant de supprimer des fichiers d'un dossier de travail.

### [Afficher l'historique des fichiers : stcmd hist](#)

Décrit la commande permettant d'afficher l'historique des révisions des fichiers.

### [Lister des fichiers : stcmd list](#)

Décrit la commande permettant de lister tous les fichiers situés dans le dossier spécifié par l'option -p.

### [Verrouiller et déverrouiller des fichiers : stcmd lck](#)

Décrit la commande permettant de verrouiller et de déverrouiller des fichiers.

[Verrouiller et déverrouiller un serveur : stcmd server-mode](#)

Décrit la commande permettant de verrouiller et de déverrouiller une configuration de serveur.

[Supprimer des fichiers : stcmd remove](#)

Décrit la commande permettant de retirer des fichiers du contrôle de version.

[Définir les options personnelles : stcmd set-personal-options](#)

Décrit la commande permettant de définir et d'afficher les options personnelles.

[Mettre à jour l'état d'un fichier : stcmd update-status](#)

Décrit la commande permettant de mettre à jour l'état d'un fichier.

## Options communes

Certaines options apparaissent dans toutes, ou presque toutes, ces commandes. Ayant exactement la même signification dans chaque commande, elles font l'objet d'une explication unique dans cette section. Les options qui n'apparaissent pas dans toutes les commandes ou dont la signification varie d'une commande à l'autre sont expliquées avec les commandes concernées.

La syntaxe de toutes les lignes de commande est interprétée comme une syntaxe UNIX plutôt que comme une syntaxe Windows.

Toutes les options peuvent être écrites avec un trait d'union ( `-` ) ou une barre oblique ( `/` ), en fonction de votre système d'exploitation. Lorsque ce chapitre indique qu'une option requiert des guillemets, leur utilisation est recommandée, bien que l'option puisse être acceptée par votre système d'exploitation sans leur présence. L'adoption de cette stratégie est à la fois cohérente et sûre.

Les systèmes d'exploitation Windows requièrent des guillemets lorsque l'option contient un espace. Par exemple, lorsqu'un commentaire de révision comprend plusieurs mots, il doit être placé entre guillemets. Un commentaire composé d'un seul mot ne requiert pas de guillemets, même si vous pouvez en utiliser. Gardez à l'esprit que les commandes envoyées via la ligne de commande Java échouent si les noms des fichiers et des dossiers de travail comprennent des espaces. Vous ne pouvez utiliser des espaces dans les noms UNIX que si vous placez ceux-ci entre guillemets doubles.

## Conventions syntaxiques

La syntaxe de la ligne de commande utilise les conventions suivantes :

**[ ]**

Crochets identifiant la syntaxe des options.

**|**

Une barre verticale sépare des options mutuellement exclusives. Vous ne sélectionnez qu'une seule option.

**-?**

Liste la syntaxe de la commande et une description de chaque option.

`-?` appelle l'aide également. `-help` et `-h` sont synonymes de `-?`. `-?` fonctionne avec chaque commande, bien que l'option n'apparaisse pas dans la syntaxe. Ces informations sont envoyées à `stderr` plutôt qu'à `stdout`. Pour capturer les informations `stderr` depuis l'invite de commandes de Windows, vous devez utiliser `"2>"` plutôt que `">"` qui capture les informations `stdout`.

**-active**

Indique l'élément de processus actif.

**-cmp**

Comprime toutes les données envoyées de la station de travail au serveur et les décompresse une fois celles-ci arrivées. Sans cette option, aucune compression n'a lieu.

La compression est essentiellement utile et appropriée lorsque le client et le serveur communiquent via une connexion lente. Pour savoir s'il convient d'utiliser la compression, vous devez évaluer le cas de test suivant : le temps supplémentaire nécessaire à la compression et à la décompression des données est-il, ou non, inférieur au temps supplémentaire qui résulterait du transfert de données non compressées via la connexion lente ?

## **-csf**

Lorsque la commande associe le dossier spécifié par l'option `-p` au dossier StarTeam sous-jacent, l'utilisation de l'option `-csf` permet à la commande de différencier les dossiers StarTeam grâce à l'utilisation des minuscules/majuscules dans leur nom. Cette option ne modifie pas la distinction minuscules/majuscules pour les noms des fichiers appartenant à ces dossiers. Par exemple, avec `-csf`, les dossiers StarTeam nommés "doc" et "Doc" sont identifiés comme étant des dossiers différents. Sans cette option, les deux dossiers sont le dossier "doc".

Par défaut, les dossiers StarTeam ne sont pas différenciés d'après la casse des lettres de leur nom.

Avec ou sans l'option `-csf`, si des noms de dossiers sont ambigus, une erreur se produit. Par exemple, lorsque vous utilisez `-csf`, il y aura ambiguïté s'il existe à la fois un dossier Doc et un dossier doc. Lorsque vous n'utilisez pas `-csf`, il y a ambiguïté si les noms des dossiers s'écrivent de manière identique.

## **-encrypt**

Chiffre toutes les données envoyées de la station de travail au serveur et les déchiffre une fois celles-ci arrivées. Sans cette option, aucun chiffrement n'a lieu. Le chiffrement empêche la lecture des fichiers et des autres informations de projet par des tiers non autorisés sur des lignes réseau non sécurisées.

La syntaxe complète est :

```
-encrypt typeChiffrement
```

Les types de chiffrement sont :

RC4 : Chiffrement en continu RSA RC4 (rapide)

RC2\_ECB : Chiffrement par bloc RSA RC2 (ECB)

RC2\_CBC : Chiffrement par bloc RSA RC2 (CBC)

RC2\_CFB : (Plates-formes Windows uniquement) Chiffrement par bloc RSA RC2 (CFB)

Ces types de chiffrement sont classés du plus rapide au plus lent. De deux types de chiffrement contigus, le plus lent est le plus sûr.

**Remarque :** Pour les plates-formes autres que Windows, les clés publique et privée utilisées dans le chiffrement ne sont pas automatiquement créées. Elles sont stockées dans un fichier dans le répertoire de base de l'utilisateur. Ce fichier d'options est nommé `.starteam`. Il contient une variable ou une variable shell appelée `keyfile`. La variable `keyfile` spécifie l'emplacement du fichier qui contient les clés publique et privée. Si vous ne spécifiez pas la variable `keyfile`, une erreur se produit. Lorsque vous spécifiez la variable `keyfile` mais que le fichier n'existe pas, le client StarTeam génère une paire aléatoire de clés, crée le fichier et y stocke les clés. Veillez à sécuriser ce fichier. Par exemple, sous UNIX, seul son propriétaire doit être en mesure de le lire.

## **-eol [on|off]**

Réalise la conversion des fins de ligne dans les fichiers texte.

## **-epwdfile**

Stocke le mot de passe d'un utilisateur sous forme d'une valeur chiffrée dans un fichier local. Cette fonctionnalité prend en charge les scripts de build automatisés, qui peuvent s'exécuter sans intervention. Le script appelle `stcmd` avec un nom d'utilisateur spécifique et le nom du fichier qui contient le mot de passe chiffré correspondant

à ce nom d'utilisateur. Le mot de passe est ensuite déchiffré en interne et transmis au serveur sans passer dans le réseau sous forme de texte lisible.

Le mot-clé `-epwdfilename` spécifie le chemin d'accès au fichier contenant le mot de passe chiffré. Comme `-pwdfilename`, `-epwdfilename` remplace le mot de passe utilisé dans l'option `-p`, ce qui empêche que d'autres voient le mot de passe de l'utilisateur sur la ligne de commande. La syntaxe complète est :

```
-epwdfilename "cheminFichier"
```

La tableau suivant donne la syntaxe des commandes qui peuvent être utilisées pour stocker un mot de passe chiffré.

Action	Syntaxe
Pour être invité à entrer le mot de passe qui sera chiffré et stocké dans un fichier.	<pre>stcmd store-password -epwdfilename "cheminFichier"</pre>
Pour inclure le mot de passe chiffré dans la commande sous forme de texte lisible. (Remarquez que cette action ne permet pas d'accéder au réseau avec la valeur lisible.)	<pre>stcmd store-password -epwdfilename "cheminFichier" -password "motDePasse"</pre>

Une fois qu'un mot de passe chiffré est stocké, d'autres commandes `stcmd` peuvent spécifier `-epwdfilename` `"cheminFichier"` comme paramètres. Par exemple :

```
stcmd delete-local -p "JMarsh:password@Orion:1024/StarDraw/StarDraw/SourceCode" -epwdfilename  
"C:\estuff\myfile.txt" -filter "N" "*"
```

## -f NCI

Applique la commande à tous les fichiers nécessitant un archivage.

`-f NCI` est ignoré si `-filter` est utilisé.

## fichiers...

Spécifie les fichiers à utiliser dans la commande, par leur nom ou par un modèle de nom de fichier tel que `"*.c"`. Toutes les options sont interprétées à l'aide des conventions sémantiques UNIX au lieu de celles de Windows car les conventions UNIX sont plus spécifiques. De ce fait, `"*"`, plutôt que `"*.*"` signifie "tous les fichiers". Le modèle `"*.*"` signifie "tous les fichiers ayant une extension dans leur nom de fichier". Par exemple, le modèle `"star*.*"` trouve `starteam.doc` et `starteam.cpp` mais pas `starteam`. Pour trouver tous ces fichiers, il aurait fallu utiliser le modèle `"star*"`.

Sans cette option, la valeur par défaut est `"*"`. Lorsque cette option est utilisée, elle doit toujours être la dernière option. Toute option qui lui succède est ignorée.

Si, pour indiquer tous les fichiers, vous utilisez `*` plutôt que `"*"`, un shell UNIX développe le modèle en une série d'éléments qu'il transmet à la commande `stcmd` sous forme d'un groupe d'options. Cela risquant de générer des problèmes (par exemple, lorsque vous extrayez des fichiers manquants), il est préférable d'utiliser `"*"` pour les éviter.

Si vous utilisez plusieurs modèles de noms de fichiers, vous devez placer chacun d'eux dans son propre jeu de guillemets. Par exemple, vous pouvez utiliser `"*.bat" "*.c"`, mais vous ne pouvez pas utiliser `"*.bat *.c"`.



**Remarque :** Nous recommandons de mettre cette option entre guillemets, quelle que soit la plate-forme, mais pour des raisons différentes. Sous les plates-formes Windows, les noms de fichiers et de dossiers qui contiennent des espaces ne sont pas correctement interprétés (sauf si vous utilisez des guillemets). Sous les plates-formes UNIX, si vous n'utilisez pas de guillemets, le shell développe l'option, puis transmet au client la liste des éléments issus du développement. Cela génère fréquemment des résultats non voulus. Vous pouvez éviter ces deux conséquences en plaçant cette option entre guillemets. Il n'est conseillé d'omettre les guillemets que s'il est essentiel que l'option soit développée par le shell UNIX. Les échecs inexplicables des scripts de commandes par lot sur les deux plates-formes peuvent provenir de cette omission.

Plusieurs caractères spéciaux peuvent être utilisés dans la spécification des noms de fichiers :

\*

Correspond à une chaîne, même vide. Par exemple, \* correspond à n'importe quel nom de fichier, avec ou sans extension. "xyz\*" correspond à "xyz" comme à "xyz.cpp" et à "xyzutyfj".

?

Correspond à un caractère. Par exemple, "a?c" correspond à "abc" mais pas à "ac".

[...]

Correspond à n'importe lequel des caractères placés entre crochets.

Une paire de caractères séparés par un trait d'union (-) spécifie une plage de caractères à respecter.

Si le premier caractère qui suit un crochet ouvrant ( [ ) est un point d'exclamation ( ! ) ou un accent circonflexe ( ^ ), les caractères qui suivent ne sont pas recherchés. Tout caractère non placé entre les crochets est recherché. Par exemple, "x[a-d]y" correspond à "xby" mais pas à "xey". "x[!a-d]y" correspond à "xey" mais pas à "xby".

Un trait d'union (-) ou un crochet fermant ( ] ) peut être recherché si vous l'incluez en tant que premier ou dernier caractère d'un ensemble entre crochets.

Pour utiliser un astérisque (\*), un point d'interrogation (?) ou un crochet ouvrant ( [ ) dans un modèle, vous devez le faire précéder du caractère d'échappement (c'est-à-dire, d'une barre oblique inverse (\)).

## **-filter**

Filtre de l'état des fichiers. Les états sont **C** = (Current) A jour, **M** = Modifié, **O** = Obsolète, **N** = (Not in view) Pas dans la vue, **I** = (missing) Manquant, **G** = (merge) Fusion et **U** = (Unknown) Inconnu. Par exemple, utiliser **CM** appliquera la commande uniquement aux fichiers dont l'état est A jour ou Modifié.

**-filter** a priorité sur **-f NCI**. Si vous utilisez **G**, **O** ou **U**, vous devez aussi spécifier **-i** ou **-o**. Sinon, **G**, **O** ou **U** est ignoré.

**-filter** a aussi priorité sur **-f NCO**. Si vous utilisez **G**, **M**, **O** ou **U**, vous devez aussi spécifier **-merge** ou **-o** pour forcer l'opération d'extraction. Sinon, **G**, **M**, **O** ou **U** est ignoré.

## **-fp**

Redéfinit le dossier de travail ou le répertoire de travail du dossier StarTeam spécifié. Cela est équivalent à définir pour le dossier un chemin de travail secondaire.

Cette option, tout en vous permettant d'utiliser un dossier de travail différent de celui spécifié par la vue StarTeam, se caractérise par sa compatibilité multiplate-forme. Par exemple, les systèmes UNIX et Windows spécifient les noms de lecteurs et de chemins d'accès aux répertoires de manière incompatible.

Le chemin `D:\MYPRODUCT\DEVELOPMENT\SOURCE` est compris par une plate-forme Windows mais ne l'est pas par une plate-forme UNIX. Utilisez cette option pour définir le chemin de travail si votre plate-forme ne comprend pas le chemin spécifié dans le projet StarTeam.

Une barre oblique inverse (`\`) est interprétée comme caractère d'échappement lorsqu'elle précède des guillemets. Par conséquent, une erreur se produit dans l'exemple suivant :

```
stcmd ci -p "xxx" -fp "C:\\" "*" 
```

qui est interprété ainsi :

```
stcmd ci -p "xxx" -fp "C:" "*" 
```

Pour éviter ce type de situation, insérez un caractère d'échappement avant le dernier caractère de l'expression `"C:\`", comme suit :

```
stcmd ci -p "xxx" -fp "C:\\\" "*" 
```

Vous pouvez également l'éviter comme suit lorsque le chemin `-rp` ne finit pas par le dossier racine comme dans l'expression `C:\orion\` :

```
stcmd ci -p "xxx" -fp "C:\orion" "*" 
```

La syntaxe complète est :

```
-rp "nomDossier"
```

"Dossier" est le terme Windows et il apparaît dans l'interface utilisateur de StarTeam. "Répertoire" est le terme correct pour la plate-forme UNIX.

## **-h**

Appelle l'aide. `-help` fonctionne avec chaque commande, bien qu'elle n'apparaisse pas dans la syntaxe.

## **-help**

Appelle l'aide. `-help` fonctionne avec chaque commande, bien qu'elle n'apparaisse pas dans la syntaxe.

## **-i**

Demande à l'utilisateur de confirmer l'archivage lorsque l'état d'un fichier est Fusion, Obsolète ou Inconnu.

## **-is**

Applique la commande à tous les dossiers enfant. Sans cette option, la commande s'applique uniquement au dossier spécifié.

Lorsque vous utilisez cette option avec l'option `add-folder`, vous pouvez ajouter une branche entière de dossiers à l'arborescence de dossiers de StarTeam. Lorsque vous l'utilisez avec l'option `add-project`, vous pouvez créer un projet non limité à un dossier racine.

## **-l**

Verrouille un fichier.

## **-mark**

Marque une demande de modification comme étant corrigée, une exigence comme étant achevée, ou une tâche comme étant terminée.

## **-nel**

Verrouille un fichier en mode non exclusif

## **-nologo**

Supprime la mention de droits d'auteur. `-nologo` fonctionne avec chaque commande, bien qu'elle n'apparaisse pas dans la syntaxe.

## **-nomove**

Ne déplace pas les étiquettes déjà attachées.

## **-o**

Force l'archivage.

## **-p**

Indique la vue ou le dossier à utiliser ; en outre, fournit le nom d'utilisateur et le mot de passe requis pour accéder au serveur. La syntaxe complète est :

```
-p "nomUtilisateur:motPasse@nomHôte:pointTerminaison/nomProjet/[nomVue/]  
[hiérarchieDossiers/]"
```

Par exemple :

```
-p "bsmith:rocketfive@orion:49201/StarDraw/StarDraw/SourceCode/"
```

- ◆ Si le nom d'utilisateur est omis, le nom d'utilisateur actuel est utilisé.
- ◆ Si le mot de passe est omis, l'utilisateur est invité à le saisir. Lorsque l'utilisateur saisit un mot de passe, les caractères ne sont pas affichés.
- ◆ Si le nom d'hôte est omis, la valeur par défaut est localhost.
- ◆ La saisie d'un point de terminaison (numéro de port) est obligatoire. La valeur par défaut est 1024.
- ◆ Le nom de projet est toujours requis.
- ◆ Une hiérarchie de vues doit être utilisée pour identifier la vue. Utilisez le signe "deux points" (:) comme délimiteur des noms de vues. La hiérarchie des vues doit toujours comprendre la vue racine. Par exemple, "StarDraw:Release 4:Service Packs" indique que la vue à utiliser est la vue Service Packs, qui est un enfant de la vue Release 4 et un petit-enfant de la vue racine StarDraw. Si le nom de vue est omis, la vue racine est

utilisée. Si seule cette vue porte ce nom dans ce projet, vous pouvez n'utiliser que le nom de la vue. Toutefois, cela n'est pas recommandé car une autre vue de même nom pourrait être créée à une date ultérieure.

- ◆ Une hiérarchie de dossiers doit être utilisée pour identifier le dossier. Utilisez la barre oblique (/) comme délimiteur des noms de dossiers. La hiérarchie de dossiers ne comprend jamais le dossier racine. Omettez la hiérarchie de dossiers si le fichier figure dans le dossier racine de la vue. Par exemple, si le dossier racine de la vue est `StarDraw` et que la hiérarchie d'accès à vos fichiers est `StarDraw/SourceCode/Client`, utilisez uniquement "SourceCode/Client".

Si l'une des variables utilisées avec cette option contient des caractères pouvant être utilisés comme délimiteurs, utilisez le signe pourcentage (%) suivi du code hexadécimal correspondant à la place de ces caractères. Par exemple, si "@" apparaît comme caractère dans un mot de passe, vous devez le remplacer par "%40".

Pour ":", utilisez "%3a"

Pour "/", utilisez "%2f"

Pour "@", utilisez "%40"

Pour "%", utilisez "%25"

Sous UNIX et d'autres systèmes d'exploitation, certains caractères spéciaux doivent être précédés d'une barre oblique inverse "\" ou d'un autre caractère d'échappement. Dans l'option `-p`, vous pouvez remplacer ces caractères par des codes hexadécimaux. Par exemple, "%3c" peut être utilisé sous UNIX à la place de "<".

Pour un espace, utilisez "%20"

Pour "<", utilisez "%3c"

Pour ">", utilisez "%3e"

## **-pwdfile**

Spécifie le chemin d'accès du fichier qui stocke le mot de passe de l'utilisateur. Cette option redéfinit le mot de passe utilisé dans l'option `-p`. Elle empêche l'affichage du mot de passe de l'utilisateur dans la ligne de commande. La syntaxe complète est :

```
-pwdfile "cheminFichier"
```

## **-q**

Supprime l'information de progression. Sans cette option, un message relatif à chaque action s'affiche parallèlement à la réalisation de celle-ci.

## **-r**

Raison de l'archivage.

## **-rf**

Précède le nom du fichier qui contient la raison de l'archivage.

## **-ro**

Définit le fichier comme étant en lecture seule après l'opération.

## **-rp**

Spécifie ou redéfinit le dossier de travail, ou répertoire de travail, du dossier racine de la vue StarTeam. La commande `stcmd add-project` utilise cette option pour spécifier le dossier de travail du dossier racine de la nouvelle vue. Les autres commandes l'utilisent pour redéfinir le dossier de travail existant.

Cette option, tout en vous permettant d'utiliser un dossier de travail différent de celui spécifié par la vue StarTeam, se caractérise par sa compatibilité multiplate-forme. Par exemple, les systèmes UNIX et Windows spécifient les noms de lecteurs et de chemins d'accès aux répertoires de manière incompatible.

Le chemin `D:\MYPRODUCT\DEVELOPMENT\SOURCE` est compris par une plate-forme Windows mais ne l'est pas par une plate-forme UNIX. Utilisez cette option pour définir le chemin de travail si votre plate-forme ne comprend pas le chemin spécifié dans le projet StarTeam.

Le shell UNIX interprète une barre oblique inverse (`\`) comme un caractère d'échappement lorsqu'elle précède certains caractères, tels que les guillemets. Par conséquent, une erreur se produit dans l'exemple suivant :

```
stcmd ci -p "xxx" -rp "C:\" "*"
```

qui est interprété ainsi :

```
stcmd ci -p "xxx" -rp "C:" "*"
```

Pour éviter ce type de situation, insérez un caractère d'échappement avant le dernier caractère de l'expression `"C:\"`, comme suit :

```
stcmd ci -p "xxx" -rp "C:\\\" "*"
```

Vous pouvez également l'éviter comme suit lorsque le chemin `-rp` ne finit pas par le dossier racine comme dans l'expression `"C:\orion\"` :

```
stcmd ci -p "xxx" -rp "C:\orion" "*"
```

La syntaxe complète est :

```
-rp "nomDossier"
```

"Dossier" est le terme Windows et il apparaît dans l'interface utilisateur de StarTeam. "Répertoire" est le terme correct pour la plate-forme UNIX.

## **-rw**

Définit le fichier comme étant en lecture et en écriture après l'opération.

## **-stop**

Cette option est souvent utilisée avec `-x`. Interrompt l'exécution de la ligne de commande dès qu'une erreur se produit. Sans cette option, l'exécution continue malgré les erreurs.

**-u**

Déverrouille un fichier.

**-v**

Etiquette de version.

**-x**

Passes du mode interactif au mode de commandes (et vice versa). Sans cette option, vous devez confirmer les messages d'erreur de façon interactive et les codes de sortie peuvent ne pas être disponibles. Avec cette option, aucun message d'erreur n'est affiché, mais des codes de sortie sont définis. Les codes de sortie sont 0 en cas de réussite et 1 en cas d'échec.

**Concepts associés**

[Caractères spéciaux](#)

**Référence associée**

[Opérations en ligne de commande du client](#)

## Caractères spéciaux

\* Correspond à une chaîne, même vide. Par exemple, "x\*z" correspond à "xyz" et à "xz". ? Correspond à un caractère. Par exemple, "a?c" correspond à "abc" mais pas à "ac".

[...] Correspond à n'importe lequel des caractères placés entre crochets.

Une paire de caractères séparés par un trait d'union (-) spécifie une plage de caractères à rechercher. Si le premier caractère qui suit un crochet gauche ( [ ) est un point d'exclamation ( ! ) ou un accent circonflexe ( ^ ), les caractères qui suivent ne sont pas recherchés. Tout caractère non placé entre les crochets est recherché. Un trait d'union ( - ) ou crochet droit ( ] ) peut être recherché si vous l'incluez en tant que premier ou dernier caractère dans un ensemble entre crochets. Par exemple, "x[a - d]y" correspond à "xby" mais pas à "xey" tandis que "x[!a - d]y" correspond à "xey" mais pas à "xby". Si vous souhaitez utiliser un astérisque (\*), un point d'interrogation (?) ou un crochet gauche ( [ ) dans un modèle, vous devez le faire précéder du caractère d'échappement (c'est-à-dire, une barre oblique inverse \ ).

Si vous utilisez \* plutôt que "\*" pour indiquer tous les fichiers, un shell UNIX en extrait une série d'éléments qu'il transmet à la commande `stcmd` sous la forme d'un groupe d'options. Cela pouvant générer des complications et des problèmes (par exemple, lorsque vous extrayez des fichiers manquants), il est préférable d'utiliser "\*". Si vous utilisez plusieurs modèles de noms de fichiers, vous devez placer chacun d'eux dans son propre jeu de guillemets. Par exemple, vous pouvez utiliser "\*.bat" "\*.c", mais vous ne pouvez pas utiliser "\*.bat \*.c".

Ces caractères spéciaux s'appliquent également à l'option `fichiers...` disponible dans certaines commandes.

### Référence associée

[Opérations en ligne de commande du client](#)

## Codes de sortie

Les commandes `stcmd` renvoient des codes de sortie si l'option `-x` est utilisée dans ces commandes. Ces codes sont : 0 pour réussite, 1 pour échec, 101 si au moins un des modèles de fichiers spécifiés n'a pas trouvé de correspondance, 102 si aucun des modèles de fichiers spécifiés n'a trouvé de correspondance. La commande `stcmd diff` possède une option supplémentaire (`-e`) qui renvoie des codes de sortie. L'option `-e` possède trois codes de sortie (0, 1 et 2) dont les significations diffèrent de celles répertoriées ci-dessus. En outre, le code 1 ne peut pas être renvoyé si vous n'utilisez pas également l'option `-x`.

## Pour les plates-formes Windows

Vous pouvez utiliser `ERRORLEVEL` dans un fichier batch pour effectuer des opérations en fonction du résultat d'une des commandes qu'il contient. Par exemple, après une commande `stcmd` dans un fichier batch, vous pouvez utiliser :

```
IF ERRORLEVEL int instruction
```

où int est 0 ou 1.

Par exemple :

```
IF NOT ERRORLEVEL 1 GOTO OPOK
ECHO ERREUR RENCONTREE A ETAPE5>LOGFIC.TXT.

:OPOK
```

Vous pouvez aussi utiliser la pseudo variable d'environnement `%ERRORLEVEL%`. Par exemple, vous pouvez utiliser ce qui suit dans un shell ou sur la ligne de commande (après une commande `stcmd`) :

```
SET /A STEPNUMBER=5
SET /A THISERROR=STEPNUMBER*ERRORLEVEL
SET /A ERRORMASK=+THISERROR
```

## Pour UNIX

Chaque shell possède sa propre méthode d'évaluation des codes de sortie. Par exemple, dans le shell Bourne, l'instruction suivante peut venir à la suite d'une commande `stcmd` :

```
if [ return ]; then instruction
```

### Référence associée

[Opérations en ligne de commande du client](#)



## Ajouter des fichiers : stcmd add

Utilisez `stcmd add` pour ajouter des fichiers à un projet depuis la ligne de commande.

Vous pouvez simultanément lier les fichiers ajoutés à un élément de processus. Tous les fichiers correctement ajoutés à l'aide de cette commande sont liés et épinglés à la dernière révision de l'élément de processus. Utilisez l'option `-active` pour spécifier l'élément de processus actuellement actif (précédemment défini à l'aide d'un client StarTeam sur votre station de travail).

Si aucun élément n'est actif ou si vous préférez en utiliser un autre, recourez à l'option indiquant le type de l'élément de processus (`-cr`, `-req` ou `-task`), suivie du chemin complet menant du dossier racine de la vue de projet StarTeam à l'élément, en utilisant la barre oblique (`/`) comme délimiteur des noms de dossiers. Pour les éléments de processus hors de vues, spécifiez le nom du projet et le nom de la vue devant le chemin complet du dossier. Séparez le chemin de la vue par un deux-points (`:`). Par exemple, `-cr MyProject/RootView:ChildView/SourceCode/37` spécifie la demande de modification 37 dans le dossier `SourceCode` de la vue `ChildView` du projet `MyProject`. Au cours de l'exécution, le processus suppose d'abord que l'élément de processus est dans la vue en cours puis vérifie la vue en cours pour déterminer si le chemin d'accès complet correspond à un chemin de dossier à l'intérieur de cette vue. Si l'élément de processus n'est pas trouvé dans la vue en cours, il est considéré comme élément de processus hors de la vue et la recherche de cet élément de processus commence à partir du projet et de la vue.

Utilisez l'option `-mark` pour marquer simultanément l'élément de processus comme étant corrigé, terminé ou achevé, suivant son type. Par exemple, une demande de modification peut être marquée comme étant corrigée. L'élément n'est marqué comme étant corrigé, terminé ou achevé que si tous les fichiers sont correctement ajoutés.

## Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd add -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf] [-encrypt
typeChiffrement]
[-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp "cheminDossier"] [- l | -u | -nel] [-ro
| -rw]] [-d "description"]
[-vl "nomEtiquette"] [-eol [on | off]] [[ -active | [-cr | -req | -task ] cheminProcessus]
[-mark]] [-short] [fichiers...]
```

Option	Description
<code>-active</code>	Indique l'élément de processus actif.
<code>-cr</code> <code>-req</code> <code>-task</code>	Chemin complet depuis le dossier racine de la vue de projet vers la demande de modification, l'exigence ou le numéro de la tâche à utiliser comme élément de processus. Utilisez la barre oblique ( <code>/</code> ) comme délimiteur des noms de dossiers.  Pour les éléments de processus hors de vues, spécifiez le nom du projet et le nom de la vue devant le chemin complet du dossier. Séparez le chemin de la vue par un deux-points ( <code>:</code> ). Par exemple, <code>-cr MyProject/RootView:ChildView/SourceCode/37</code> spécifie la demande de modification 37 dans le dossier <code>SourceCode</code> de la vue <code>ChildView</code> du projet <code>MyProject</code> .  Au cours de l'exécution, le processus suppose d'abord que l'élément de processus est dans la vue en cours puis vérifie la vue en cours pour déterminer si le chemin d'accès complet correspond à un chemin de dossier à l'intérieur de cette vue. Si l'élément de processus n'est pas trouvé dans la vue en cours, il est considéré comme élément de processus hors de la vue et la recherche de cet élément de processus commence à partir du projet et de la vue.
<code>-d</code>	Définit la description des nouveaux fichiers ajoutés en utilisant la description spécifiée. Cette description est entourée de guillemets.
<code>-eol</code>	Convertit automatiquement les marques de fin de ligne en retour chariot/saut de ligne lorsque les fichiers texte de travail sont transférés vers le référentiel du serveur. Lorsque cette option est activée, la marque de fin de ligne en

cours dans le fichier de travail est convertie en une combinaison retour chariot/saut de ligne. Lorsque cette option est désactivée (par défaut), aucune conversion de fin de ligne n'est exécutée.

Définissez cette option par on, par exemple, lorsque vous ajoutez au référentiel un fichier de travail issu d'un ordinateur sous UNIX, si vous voulez que le référentiel stocke les fichiers texte avec un retour chariot/saut de ligne.

<code>-l</code>	Verrouille chaque fichier après son ajout à la vue. Sans <code>-l</code> , <code>-u</code> ni <code>-nel</code> , les fichiers sont par défaut déverrouillés.
<code>-mark</code>	Indique que, si tous les fichiers sont correctement ajoutés, l'état de l'élément de processus passe à Corrigé (pour une demande de modification), Terminé (pour une tâche) ou Achevé (pour une exigence). Les fichiers sont épinglés à la révision avec le nouvel état.
<code>-nel</code>	Verrouille chaque fichier en mode non exclusif après son ajout.
<code>-ro</code>	Rend le fichier de travail accessible uniquement en lecture après cette opération. Sans cette option, le fichier reste tel qu'il était avant l'opération. Habituellement, vous utiliserez <code>-ro</code> pour vous éviter de modifier un fichier non verrouillé par vous.  L'option <code>-ro</code> doit être utilisée avec <code>-l</code> , <code>-u</code> , ou <code>-nel</code> . Si vous utilisez <code>-ro</code> , vous ne pouvez pas utiliser <code>-rw</code> .
<code>-rw</code>	Rend le fichier de travail accessible en lecture et en écriture après cette opération. Sans cette option, le fichier reste tel qu'il était avant l'opération.  L'option <code>-rw</code> doit être utilisée avec <code>-l</code> , <code>-u</code> , ou <code>-nel</code> . Si vous utilisez <code>-rw</code> , vous ne pouvez pas utiliser <code>-ro</code> .
<code>-u</code>	Laisse les nouveaux fichiers ajoutés déverrouillés.
<code>-vl</code>	Spécifie une étiquette à appliquer aux nouveaux fichiers. Cette étiquette est entourée de guillemets. Cette option peut apparaître plusieurs fois dans la commande. L'étiquette peut être une étiquette de vue ou de révision, mais elle doit déjà exister dans l'application.

## Exemple

L'exemple suivant utilise la syntaxe `stcmd add` pour ajouter tous les fichiers `.doc` ayant pour état Pas dans la vue à User Manual, enfant du dossier racine StarDraw (dans la vue StarDraw du projet StarDraw). Il verrouille les fichiers et leur donne la description "Premier jet du chapitre".

```
stcmd add -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/User Manual" -l-d "Premier jet du chapitre" "*.doc"
```

## Référence associée

[Opérations en ligne de commande du client](#)

## Ajouter des dossiers : stcmd add-folder

Utilisez `stcmd add-folder` pour ajouter des dossiers StarTeam à une vue depuis la ligne de commande. Vous pouvez ajouter le dossier au dossier racine ou à n'importe quel autre dossier appartenant à cette vue. Le dossier de travail de votre nouveau dossier StarTeam est créé par défaut dans StarTeam, et non sur votre station de travail. Le dossier de travail porte le même nom que le dossier StarTeam. Il s'agit d'un dossier enfant du dossier de travail pour le parent du dossier StarTeam.

Par exemple, supposons que vous créez un dossier StarTeam nommé "Wizard". Wizard est un enfant d'un dossier StarTeam dont le dossier de travail est "C:\StarDraw". Par conséquent, le dossier de travail de Wizard devient "C:\StarDraw\Wizard".

L'utilisation de l'option `-is` vous permet d'ajouter une branche de dossiers à l'arborescence des dossiers de la vue de projet. Lorsque vous utilisez `-is`, recourez à `-rp` ou à `-fp` pour spécifier le dossier, sur votre station de travail, dont les dossiers enfant deviennent les dossiers enfant du nouveau dossier StarTeam. L'utilisation de l'option `-fp` est recommandée, car elle spécifie le chemin d'accès direct au parent de ces dossiers enfant. En revanche, l'option `-rp`, qui spécifie le chemin du dossier de travail utilisé pour le dossier racine de la vue, ajoute les noms des dossiers StarTeam, situés dans l'arborescence entre le dossier racine et le nouveau dossier, à la fin du chemin que vous spécifiez. Les options `-rp` et `-fp` n'ont d'effet sur cette commande que lorsque vous utilisez l'option `-is`.

### Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd add-folder -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf] [-encrypt typeChiffrement] [-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp "cheminDossier"] -name "nomDossier" [-d "description"] [-ex "typeExclusion"] [-exlist "masqueFichier" | -exfile "nomFichier"]
```

Option	Description
<code>-d</code>	Spécifie une description pour le dossier. Utilisez au maximum 254 caractères.
<code>-ex</code>	<p>Indique les listes d'exclusion à utiliser pour ce nouveau dossier. Les listes d'exclusion permettent d'exclure certains fichiers ou types de fichiers de l'affichage. Si un fichier de travail se trouvant dans le dossier de travail de ce dossier, a l'état Pas dans la vue mais correspond à une spécification de fichier indiquée dans l'une des listes d'exclusion, l'application ne l'affiche pas. C'est comme si le fichier n'existait pas.</p> <p>Par exemple, supposons que vous créez des fichiers dans une application qui réalise des copies automatiques de sauvegarde (d'extension <code>.bak</code>) chaque fois que vous enregistrez un fichier. Votre dossier de travail peut contenir plusieurs fichiers <code>.bak</code>, mais il n'y a pas de raison que vous les ajoutiez à la vue de projet. Dans l'application, il est gênant de voir ces fichiers <code>.bak</code> comme des candidats possibles ; nous vous conseillons donc de les exclure. L'exclusion de fichiers est réalisée dossier par dossier. Toutefois, des listes d'exclusion peuvent être héritées de dossiers parent.</p> <p>La syntaxe complète est : <code>-ex typeExclusion</code></p> <p>Les types sont :</p> <p><code>inherit</code>, qui indique que ce dossier héritera de toutes les listes d'exclusion utilisées par son dossier parent et utilisera la liste d'exclusion spécifiée avec l'option <code>-exfile</code> ou <code>-exlist</code> (le cas échéant). Il s'agit du type par défaut.</p> <p><code>local</code>, qui indique que ce dossier utilisera seulement la liste d'exclusion spécifiée avec <code>-exfile</code> ou <code>-exlist</code>.</p> <p><code>none</code>, qui indique que ce dossier n'utilisera aucune liste d'exclusion, quelles que soient les listes spécifiées avec <code>-exfile</code> ou <code>-exlist</code>.</p>
<code>-exfile</code>	Spécifie le chemin du fichier qui contient la liste d'exclusion locale de ce dossier. Voir <code>-exlist</code> pour obtenir une description du contenu de la liste d'exclusion.

---

**-exlist** Spécifie la liste d'exclusion locale de ce dossier. Utilisez au maximum 254 caractères. Entrez une ou plusieurs spécifications de fichier (avec les caractères génériques standard \* et ?) séparées par des virgules, des espaces ou des points-virgules. Pour intégrer une virgule, un espace ou un point-virgule dans la spécification, indiquez-le entre guillemets.

Par exemple : \*.exe,\*.dll p\*z.doc;\*.t?t "test \*.\*"

Si vous utilisez des guillemets dans votre liste d'exclusion ou si votre liste est longue, nous vous recommandons d'utiliser l'option -exfile. Avec l'option -exlist, chaque guillemet dans la liste d'exclusion doit être précédé du caractère d'échappement de votre système ou de votre shell. Par exemple, l'accent circonflexe (^) fonctionne sur les systèmes NT. Avec -exfile, il est inutile d'utiliser des caractères d'échappement.

---

**-name** Spécifie un nom pour le dossier. Utilisez au maximum 254 caractères. Dans un fichier, si la liste d'exclusion contient des guillemets, le caractère d'échappement est inutile.

---

## Exemple

L'exemple suivant utilise la syntaxe `stcmd add-folder` pour créer un dossier nommé Wizard comme enfant du dossier StarDraw, dossier racine de la vue de projet StarDraw. En outre, il définit une liste d'exclusion locale pour Wizard. Par défaut, Wizard hérite des listes d'exclusion de son dossier parent et utilise en outre la liste d'exclusion locale.

```
stcmd add-folder -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/" -name "Wizard" -d "Expert de configuration de StarDraw" -exlist "*.bak"
```

L'exemple suivant crée le même dossier que l'exemple précédent. Toutefois, il inclut des dossiers enfant. Dans ce cas, le dossier ayant pour chemin "C:\Wizard" possède des dossiers enfant (Source, Spec et Doc) qui sont tous ajoutés comme dossiers StarTeam en plus de Wizard. Tous les nouveaux dossiers (Wizard, Source, Spec et Doc) possèdent les dossiers de travail par défaut que le serveur StarTeam leur affecte automatiquement, quelle que soit la valeur de l'option `-fp`. Wizard est le parent de Source, de Spec et de Doc. StarDraw est le parent de Wizard.

```
stcmd add p "JMarsh:motPasse@Orion:1024/StarDraw/ StarDraw/" -name "Wizard" -d "Expert de configuration de StarDraw" -is -fp "C:\Wizard" exlist "*.bak"
```

## Référence associée

[Opérations en ligne de commande du client](#)

## Ajouter des projets : stcmd add-project

Utilisez `stcmd add-project` pour ajouter un projet à une configuration de serveur StarTeam depuis la ligne de commande. Lorsqu'un projet est créé, sa vue racine et le dossier racine de celle-ci sont également créés. Dans cette commande, l'option `-rp` spécifie le dossier de travail de ce dossier racine.

L'option `-is` vous permet d'utiliser les dossiers enfant du dossier de travail comme dossiers enfant du dossier racine dans l'arborescence de dossiers StarTeam.

### Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd add-project [-pwdfile "cheminFichier" ] [-cmp] [-encrypt typeChiffrement] [-is] [-q]
[-x]
[-stop] -s "nomServeur" -name "nomProjet" -rp "cheminDossier" [-d "description"]
[-kw "masqueFichier" ]-kwfile "nomFichier"] [-ex "typeExclusion" ] [-exlist "masqueFichier"
]-exfile "nomFichier"]
```

Option	Description
<code>-d</code>	Spécifie une description pour le dossier. Utilisez au maximum 254 caractères.
<code>-ex</code>	Indique les listes d'exclusion à utiliser pour le dossier racine du projet.  Les listes d'exclusion permettent d'exclure certains fichiers ou types de fichiers de l'affichage. Si un fichier de travail se trouvant dans le dossier de travail de ce dossier, a l'état Pas dans la vue mais correspond à une spécification de fichier indiquée dans l'une des listes d'exclusion, l'application ne l'affiche pas. C'est comme si le fichier n'existait pas.  Par exemple, supposons que vous créez des fichiers dans une application qui réalise des copies automatiques de sauvegarde (d'extension <code>.bak</code> ) chaque fois que vous enregistrez un fichier. Votre dossier de travail peut contenir plusieurs fichiers <code>.bak</code> , mais il n'y a pas de raison que vous les ajoutiez à la vue de projet. Dans l'application, il est gênant de voir ces fichiers <code>.bak</code> comme des candidats possibles ; nous vous conseillons donc de les exclure. L'exclusion de fichiers est réalisée dossier par dossier. Toutefois, des listes d'exclusion peuvent être héritées de dossiers parent. La syntaxe complète est :  <code>-ex typeExclusion</code>  Les types sont :  <code>inherit</code> Indique que ce dossier héritera de toutes les listes d'exclusion utilisées par son dossier parent et utilisera la liste d'exclusion spécifiée avec l'option <code>-exfile</code> ou <code>-exlist</code> (le cas échéant). Il s'agit du type par défaut, même si le dossier racine n'a rien à hériter.  <code>local</code> Indique que ce dossier utilisera seulement la liste d'exclusion spécifiée avec <code>-exfile</code> ou <code>-exlist</code> .  <code>none</code> Indique que le dossier racine n'utilisera aucune liste d'exclusion, quelles que soient les listes spécifiées avec <code>-exfile</code> ou <code>-exlist</code> .
<code>-exfile</code>	Spécifie le chemin du fichier qui contient la liste d'exclusion locale du dossier racine. Voir <code>-exlist</code> pour obtenir une description du contenu de la liste d'exclusion.
<code>-exlist</code>	Spécifie la liste d'exclusion locale du dossier racine. Utilisez au maximum 254 caractères. Entrez une ou plusieurs spécifications de fichier (avec les caractères génériques standard <code>*</code> et <code>?</code> ) séparées par des virgules, des espaces ou des points-virgules. Pour intégrer une virgule, un espace ou un point-virgule dans la spécification, indiquez-le entre guillemets. Par exemple :  <code>*.exe,*.dll p*z.doc;*.t?t "test *.*"</code>  Si vous utilisez des guillemets dans votre liste d'exclusion ou si votre liste est longue, nous vous recommandons d'utiliser l'option <code>-exfile</code> . Avec l'option <code>-exlist</code> , chaque guillemet dans la liste d'exclusion doit être précédé du

caractère d'échappement de votre système ou de votre shell. Par exemple, l'accent circonflexe (^) fonctionne sur les systèmes NT. Avec -exfile, il est inutile d'utiliser des caractères d'échappement.

---

**-kw** Spécifie les extensions de fichier avec lesquelles vous voulez utiliser des mots-clés. Utilisez au maximum 254 caractères. Entrez une ou plusieurs spécifications de fichier (avec les caractères génériques standard \* et ?) séparées par des virgules, des espaces ou des points-virgules. Pour intégrer une virgule, un espace ou un point-virgule dans la spécification, indiquez-le entre guillemets. Par exemple :

```
*.cpp,*.h p*z.doc;*.t?t "test *.*"
```

Si vous utilisez des guillemets dans votre liste de mots clés ou si votre liste est longue, nous vous recommandons d'utiliser l'option -kwfile. Avec l'option -kwlist, chaque guillemet dans la liste de mots clés doit être précédé du caractère d'échappement de votre système ou de votre shell. Par exemple, l'accent circonflexe (^) fonctionne sur les systèmes NT. Avec -kwfile, il est inutile d'utiliser des caractères d'échappement.

---

**-kwfile** Spécifie le chemin du fichier contenant les extensions de fichier avec lesquelles vous voulez utiliser des mots-clés.

---

**-name** Spécifie un nom pour le dossier. Utilisez au maximum 254 caractères.

---

**-s** Identifie le serveur. La syntaxe complète est :

```
-s "nomUtilisateur:motPasse@hôte:numéroPort"
```

Par exemple :

```
-s "JMarsh:motPasse@orion:49201"
```

Si le nom d'utilisateur est omis, le nom d'utilisateur actuel est utilisé. Le nom d'utilisateur spécifié dans l'exemple est "JMarsh"

Si le mot de passe est omis, l'utilisateur est invité à le saisir. Le mot de passe spécifié dans l'exemple est "motPasse". Si le nom d'hôte est omis, la valeur par défaut est localhost. Le nom d'hôte spécifié dans l'exemple est "orion".

Le numéro de port est requis. L'exemple utilise le numéro de port par défaut : 49201.

---

## Exemple

L'exemple suivant utilise `stcmd add-project` pour créer un projet nommé Integrations sur l'ordinateur nommé Orion. (Orion exécute une instance du serveur StarTeam avec une configuration de serveur qui utilise le port 1024.) Cette commande crée le projet, spécifie que les données envoyées entre les stations de travail et le serveur doivent être compressées et chiffrées, puis attribue une description au projet.

```
stcmd add-project -s "JMarsh:motPasse@Orion:1024" -cmp -encrypt "RC4" -name "Integrations"
-rp "C:\integrations" -d "intégrations entre nos produits et les produits de nos
partenaires"
```

## Référence associée

[Opérations en ligne de commande du client](#)

## Ajouter des vues : stcmd add-view

Utilisez `stcmd add-view` pour ajouter une vue à une configuration de serveur StarTeam depuis la ligne de commande. Lorsque la vue est créée, sa vue parent est la vue spécifiée avec l'option `-p` et son dossier racine est le dossier spécifié avec l'option `-p`. Dans cette commande, l'option `-rp` spécifie le dossier de travail du dossier racine. Utilisez les options ci-après pour créer les types de vues suivants :

- ◆ Utilisez `-dr` pour créer une vue de référence accessible en lecture et en écriture.
- ◆ Utilisez `-dr -ro` pour créer une vue de référence accessible en lecture seule.
- ◆ Utilisez `-dr -ba` pour créer une vue de branchement dans laquelle le comportement des éléments existants est défini pour créer une branche en cas de modification.
- ◆ Utilisez `-dr -bn` pour créer une vue de branchement dans laquelle le comportement des éléments existants n'est pas défini pour créer une branche en cas de modification.
- ◆ Si vous n'utilisez pas `-dr`, une vue vierge est créée.

## Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd add-view -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-encrypt  
typeChiffrement]  
[-q] [-x] [-stop] -name "nomVue" [-rp "cheminDossier"] [-d "description"] [-dr [-ro | -ba |  
-bn  
[-cfigl "nomEtiquette" | -cfgp "nomEtat" | -cfgd "depuisDate"]]]
```

Option	Description
<code>-ba</code>	Utilisée avec l'option <code>-dr</code> , spécifie une vue de branchement dans laquelle le comportement des éléments existants est défini de manière à créer une branche en cas de modification. La valeur de la propriété Activer la création de branches lors de modifications pour les éléments partagés de la vue est initialement activée.  Cette option doit être utilisée avec <code>-dr</code> .
<code>-bn</code>	Utilisée avec l'option <code>-dr</code> , spécifie une vue de branchement dans laquelle le comportement des éléments existants n'est pas défini de manière à créer une branche en cas de modification. La valeur de la propriété Activer la création de branches lors de modifications pour les éléments partagés de la vue est initialement désactivée. Cette option doit être utilisée avec <code>-dr</code> .
<code>-cfgd</code>	Configure la vue à partir de la date et de l'heure spécifiées. Voici des exemples :  "12/29/01 10:52 AM"  "December 29, 2001 10:52:00 AM PST"  "Monday, December 29, 2001 10:52:00 AM PST". Cette option doit être utilisée avec l'une des combinaisons suivantes : <code>-dr -ro</code> , <code>-dr -ba</code> ou <code>-dr -bn</code> .
<code>-cfigl</code>	Configure la vue en utilisant l'étiquette spécifiée. Sans <code>-cfigl</code> , ni <code>-cfgp</code> , ni <code>-cfgd</code> , la configuration actuelle de la vue est utilisée. Cette option doit être utilisée avec l'une des combinaisons suivantes : <code>-dr -ro</code> , <code>-dr -ba</code> ou <code>-dr -bn</code> .
<code>-cfgp</code>	Configure la vue en utilisant l'état de promotion spécifié. Cette option doit être utilisée avec l'une des combinaisons suivantes : <code>-dr -ro</code> , <code>-dr -ba</code> ou <code>-dr -bn</code> .
<code>-d</code>	Spécifie une description pour la vue. Utilisez au maximum 254 caractères.

---

<code>-dr</code>	Spécifie une vue dérivée. Toutes les vues à l'exception des vues vierges sont dérivées. Voir aussi <code>-ba</code> , <code>-bn</code> et <code>-ro</code> .  Lorsqu'elle est utilisée sans <code>-ba</code> , <code>-bn</code> ni <code>-ro</code> , une vue de référence accessible en lecture et en écriture est créée. La configuration d'une telle vue est identique à celle de sa vue parent. Par conséquent, l'utilisation de <code>-dr</code> sans <code>-ba</code> , <code>-bn</code> ni <code>-ro</code> , mais avec <code>-cfigl</code> , <code>-cfigp</code> ou <code>-cfigd</code> affiche un message d'erreur.  Si cette option n'est pas utilisée, une vue vierge est créée. Pour les vues vierges, la valeur de la propriété Activer la création de branches lors de modifications pour les éléments partagés de la vue est initialement désactivée.
<code>-name</code>	Spécifie un nom pour la vue. Utilisez au maximum 254 caractères.
<code>-ro</code>	Utilisée avec <code>-dr</code> , spécifie une vue de référence accessible uniquement en lecture.

---

## Exemples

L'exemple suivant utilise `stcmd add-view` pour créer une vue de branchement nommée Maintenance 5.1 sur l'ordinateur Orion. (Orion exécute une instance du serveur StarTeam avec une configuration de serveur qui utilise le port 1024.)

Cette commande crée la vue en tant qu'enfant de la vue StarDraw existante et utilise le dossier StarDraw comme dossier racine. La nouvelle vue est basée sur l'étiquette utilisée pour le dernier build du produit 5.1 avant sa diffusion (Build 403). Elle possède un dossier de travail différent de celui du parent. Le comportement de tous les éléments appartenant à la vue est défini de manière à ce qu'une branche soit créée en cas de modification.

```
stcmd add-view -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/" -cmp -encrypt "RC4" -name
"Maintenance 5.1" -rp "C:\StarDraw\Maintenance 5.1" -d "Vue de maintenance pour la release
5.1 de notre produit"
-dr -ba -cfigl "Build 403"
```

L'exemple suivant utilise `stcmd add-view` pour créer une vue de référence accessible en lecture/écriture nommée Enracinée à SourceCode sur l'ordinateur Orion. Cette commande crée la vue en tant qu'enfant de la vue StarDraw existante et utilise le dossier SourceCode comme dossier racine. Elle possède le même dossier de travail que son parent. Etant donné qu'une vue de référence accessible en lecture/écriture doit posséder la même configuration que son parent, aucune des options `-cfigl`, `-cfigp` et `-cfigd` ne peut être utilisée.

```
stcmd add-view -p "JMarsh:motPasse@Orion:1024/ StarDraw/StarDraw/Source Code"
-cmp -encrypt "RC4" -name "Enracinée à SourceCode" -d
"Vue principale StarDraw mais avec le dossier SourceCode comme racine de l'arborescence" -
dr
```

## Référence associée

[Opérations en ligne de commande du client](#)



## Appliquer des étiquettes : stcmd apply-label

Utilisez `stcmd apply-label` pour étiqueter les révisions de fichiers spécifiées par des étiquettes de vue ou de révision. Les étiquettes doivent déjà exister dans StarTeam. Vous pouvez créer les étiquettes dans StarTeam ou en utilisant la commande `stcmd label`.

### Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd apply-label -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf]
[-encrypt typeChiffrement] [-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp
"cheminDossier"]
[-filter "étatFichier"] [-vl "nomEtiquette" | -vd "depuisDate" | -vn numéroRévision] -lbl
"nomEtiquette" [fichiers...]
```

Option	Description
<code>-filter</code>	Spécifie une chaîne d'un ou de plusieurs caractères, chacun d'entre eux représentant un état de fichier. N'incluez jamais d'espace ni aucune autre forme d'espace vierge dans cette chaîne. L'étiquette n'est appliquée qu'aux fichiers qui ont actuellement les états spécifiés. Vous ne pouvez pas appliquer d'étiquette aux fichiers dont l'état est Pas dans la vue. Les lettres utilisées pour représenter les états sont les suivantes : C = (Current) A jour, M = Modifié, G = (merGe) Fusion, O = Obsolète, I = (missing) Manquant et U = (Unknown) Inconnu.
<code>-lbl</code>	Spécifie le nom de l'étiquette à ajouter aux révisions spécifiées. Cette option peut être utilisée plusieurs fois. L'application attache toutes les étiquettes aux fichiers ou aux révisions spécifiés.
<code>-vd</code>	Spécifie la date/heure utilisée pour identifier les révisions auxquelles la nouvelle étiquette est attachée. Voici des exemples : "29/12/01 12:41" "29 décembre 2001 12:41" "Monday, December 29, 2001 12:41"
<code>-vl</code>	Spécifie l'étiquette de révision ou de vue utilisée pour identifier les révisions auxquelles la nouvelle étiquette est attachée. Cette étiquette doit déjà exister dans l'application. Sans l'option <code>-vn</code> , ou <code>-vd</code> , ou <code>-vl</code> , c'est la dernière révision de chaque fichier qui reçoit l'étiquette.
<code>-vn</code>	Spécifie le numéro de révision utilisé pour identifier les révisions auxquelles la nouvelle étiquette est attachée.

### Exemples

L'exemple suivant utilise `stcmd apply-label` pour appliquer l'étiquette Beta aux fichiers situés dans User Manual, enfant du dossier racine StarDraw (dans la vue StarDraw du projet StarDraw). StarTeam applique l'étiquette aux révisions des fichiers qui étaient à jour le 7 juillet 2003 à midi.

```
stcmd apply-label -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/User Manual" -vd
"07/07/03 12:00 PM" -lbl "Beta" "*" "
```

### Référence associée

[Opérations en ligne de commande du client](#)

## Archiver des fichiers : stcmd ci

Utilisez `stcmd ci` pour archiver des fichiers dans un référentiel (ou espace de stockage) StarTeam à partir d'un dossier de travail depuis la ligne de commande.

Vous pouvez simultanément lier les nouvelles révisions de fichier à un élément de processus. Tous les fichiers correctement ajoutés à l'aide de cette commande sont liés et épinglés à la dernière révision de l'élément de processus. Utilisez l'option `-active` pour spécifier l'élément de processus actuellement actif (précédemment défini à l'aide d'un client StarTeam sur votre station de travail).

Si aucun élément n'est actif ou si vous préférez en utiliser un autre, recourez à l'option indiquant le type de l'élément de processus (`-cr`, `-req` ou `-task`), suivie du chemin complet menant du dossier racine de la vue de projet StarTeam à l'élément, en utilisant la barre oblique (`/`) comme délimiteur des noms de dossiers. Pour les éléments de processus hors de vues, spécifiez le nom du projet et le nom de la vue devant le chemin complet du dossier. Séparez le chemin de la vue par un deux-points (`:`). Par exemple, `-cr MyProject/RootView:ChildView/SourceCode/37` spécifie la demande de modification 37 dans le dossier `SourceCode` de la vue `ChildView` du projet `MyProject`. Au cours de l'exécution, le processus suppose d'abord que l'élément de processus est dans la vue en cours puis vérifie la vue en cours pour déterminer si le chemin d'accès complet correspond à un chemin de dossier à l'intérieur de cette vue. Si l'élément de processus n'est pas trouvé dans la vue en cours, il est considéré comme élément de processus hors de la vue et la recherche de cet élément de processus commence à partir du projet et de la vue.

Utilisez l'option `-mark` pour marquer simultanément l'élément de processus comme étant corrigé, terminé ou achevé, suivant son type. Par exemple, une demande de modification peut être marquée comme étant corrigée. L'élément n'est marqué comme étant corrigé, terminé ou achevé que si tous les fichiers sont correctement ajoutés.

## Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd ci -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf] [-encrypt
typeChiffrement]
[-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp "cheminDossier"] [-filter "étatFichier"]
[-l | -u | -nel]
[-ro | -rw]] [-vl "nomEtiquette"] [-nomove] [-f NCI] [-o | -i ] [-r "commentaire" | -rf "
nomFichier"] [-eol
[on | off]] [[ -active | [-cr | -req | -task ] cheminProcessus] [-mark]] [fichiers...]
```

Option	Description
<code>-active</code>	Indique l'élément de processus actif.
<code>-cr</code>	Chemin complet depuis le dossier racine de la vue de projet vers la demande de modification, l'exigence ou le numéro de la tâche à utiliser comme élément de processus. Utilisez la barre oblique ( <code>/</code> ) comme délimiteur des noms de dossiers.
<code>-req</code>	
<code>-task</code>	Pour les éléments de processus hors de vues, spécifiez le nom du projet et le nom de la vue devant le chemin complet du dossier. Séparez le chemin de la vue par un deux-points ( <code>:</code> ). Par exemple, <code>-cr MyProject/RootView:ChildView/SourceCode/37</code> spécifie la demande de modification 37 dans le dossier <code>SourceCode</code> de la vue <code>ChildView</code> du projet <code>MyProject</code> .  Au cours de l'exécution, le processus suppose d'abord que l'élément de processus est dans la vue en cours puis vérifie la vue en cours pour déterminer si le chemin d'accès complet correspond à un chemin de dossier à l'intérieur de cette vue. Si l'élément de processus n'est pas trouvé dans la vue en cours, il est considéré comme élément de processus hors de la vue et la recherche de cet élément de processus commence à partir du projet et de la vue.
<code>-eol</code>	Convertit automatiquement les marques de fin de ligne en retour chariot/saut de ligne lorsque les fichiers texte de travail sont transférés vers le référentiel du serveur. Lorsque cette option est activée, la marque de fin de ligne en cours dans le fichier de travail est convertie en une combinaison retour chariot/saut de ligne. Lorsque cette option

est désactivée (par défaut), aucune conversion de fin de ligne n'est exécutée. Pour les clients Windows, la marque de fin de ligne est une combinaison retour chariot/saut de ligne ; pour les plates-formes UNIX, c'est un saut de ligne.

Définissez cette option par on, par exemple, lorsque vous archivez dans le référentiel un fichier de travail issu d'un ordinateur sous UNIX, si le référentiel stocke les fichiers texte avec retour chariot/saut de ligne.

- 
- `-f NCI` Spécifie l'archivage de tous les fichiers dont l'état est Modifié. NCI est l'abréviation de "needs check-in", c'est-à-dire "à archiver". Aucun autre type de fichiers n'est sélectionné pour l'archivage.
- `-f NCI` est ignoré si `-filter` est utilisé.
- 
- `-filter` Spécifie une chaîne d'un ou de plusieurs caractères, chacun d'entre eux représentant un état de fichier. N'incluez jamais d'espace ni aucune autre forme d'espace vierge dans cette chaîne. Seuls les fichiers ayant actuellement les états spécifiés seront archivés. Vous ne pouvez pas archiver les fichiers dont l'état est Pas dans la vue.
- Les lettres utilisées pour représenter les états sont les suivantes :
- C = (Current) A jour, M = Modifié, G = (merGe) Fusion, O = Obsolète, I = (missing) Manquant et U = (Unknown) Inconnu.
- `-filter` a priorité sur `-f NCI`. Si vous utilisez G, M, O ou U, vous devez aussi spécifier `-i` ou `-o`. Si vous ne le faites pas, l'option G, O ou U sera ignorée.
- 
- `-i` Permet un archivage interactif de fichiers dont l'état n'aurait normalement pas permis l'archivage. Le système vous pose la question pour chaque fichier ayant l'état Fusion, Obsolète ou Inconnu. Votre réponse peut forcer l'archivage du fichier.
- Si vous utilisez l'option `-i`, vous ne pouvez pas utiliser `-o`.
- 
- `-l` Verrouille chaque fichier après son archivage. Sans `-l,-u`, ni `-nel`, l'état de verrouillage des fichiers reste inchangé.
- 
- `-mark` Indique que, si tous les fichiers sont correctement ajoutés, l'état de l'élément de processus passe à Corrigé (pour une demande de modification), Terminé (pour une tâche) ou Achevé (pour une exigence). Les fichiers sont épinglés à la révision avec le nouvel état.
- 
- `-nel` Verrouille chaque fichier en mode non exclusif après son archivage.
- 
- `-nomove` Interrompt l'application de l'étiquette spécifiée par l'option `-vl` si le fichier en cours d'archivage a déjà une révision avec cette étiquette. Sinon, l'étiquette est déplacée de la révision actuellement étiquetée vers la nouvelle révision archivée.
- 
- `-o` Force l'archivage de fichiers dont l'état n'aurait normalement pas permis l'archivage. Cette option force l'archivage de tous les fichiers ayant l'état Fusion, Obsolète ou Inconnu.
- Si vous utilisez l'option `-o`, vous ne pouvez pas utiliser `-i`.
- 
- `-r` Fournit un commentaire de révision, qui explique généralement le motif de l'archivage des fichiers. Si vous utilisez l'option `-r`, vous ne pouvez pas utiliser `-rf`.
- 
- `-rf` Fournit le chemin du dossier qui contient le commentaire de révision.
- Si vous utilisez l'option `-rf`, vous ne pouvez pas utiliser `-r`.
- 
- `-ro` Rend le fichier de travail accessible uniquement en lecture après cette opération. Sans cette option, le fichier reste tel qu'il était avant l'opération.
- Habituellement, vous utiliserez `-ro` pour vous éviter de modifier un fichier non verrouillé par vous.
- L'option `-ro` doit être utilisée avec `-l`, `-u`, ou `-nel`. Si vous utilisez `-ro`, vous ne pouvez pas utiliser `-rw`.
- 
- `-rw` Rend le fichier de travail accessible en lecture et en écriture après cette opération. Sans cette option, le fichier reste tel qu'il était avant l'opération.
- L'option `-rw` doit être utilisée avec `-l`, `-u`, ou `-nel`. Si vous utilisez `-rw`, vous ne pouvez pas utiliser `-ro`.
- 
- `-u` Déverrouille les nouveaux fichiers archivés.

---

`-vl` Spécifie une étiquette (créée avec `stcmd label`) à appliquer aux fichiers archivés. Cette étiquette est entourée de guillemets. Cette option peut apparaître plusieurs fois dans la commande. L'étiquette peut être une étiquette de vue ou de révision, mais elle doit déjà exister dans l'application.

---

## Exemple

L'exemple suivant utilise `stcmd ci` pour archiver des fichiers `.bmp` dans Online Help, enfant du dossier racine StarDraw (dans la vue StarDraw du projet StarDraw). La commande déverrouille les fichiers, rend la copie de travail accessible uniquement en lecture et attribue aux fichiers un commentaire de révision (généralement un motif pour l'archivage des fichiers).

```
stcmd ci -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/SourceCode/Online Help" -u -ro -r  
"révisé pour la version bêta" "*.bmp"
```

## Référence associée

[Opérations en ligne de commande du client](#)

## Extraire des fichiers : stcmd co

Utilisez `stcmd co` pour extraire des fichiers d'un référentiel (ou espace de stockage) StarTeam dans votre dossier de travail à l'aide de la ligne de commande. Sauf si vous utilisez l'option `-o`, cette commande marque une pause à chaque fichier ayant pour état Modifié, Fusion ou Inconnu afin de vous indiquer qu'il ne sera pas extrait.

L'option `-merge` vous permet de fusionner les fichiers lors de l'extraction. La fusion ne fait pas partie du processus d'archivage.

### Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd co -p "projet" [-pwdfile "nomfichier"] [-epwdfile "nomfichier"] [-cmp] [-encrypt RC4, RC2_ECB, RC2_CBC, RC2_CFB] [-cfgl "étiquette" | -cfgp "état de promotion" | -cfgd "date"] [-is] [-csf] [-q] [-x] [-stop] [-rp "répertoire" | -fp "répertoire"] [-filter "filtre"][-?] [-h] [nologo] [-o | -i | -merge] [-hook "exécutable"] [-l | -u | -nel] [-ro | -rw]] [-vl "nom" | -vd "date" | -vn number] [-f NCO] [-ts] [-eol [on | off | cr| lf| crlf ]] [-fs] [-dryrun | -alwaysprompt | -neverprompt | -conflictprompt] [-mpxCacheAgent number] [-useMPXCacheAgent "hôte"port" | autolocate]] [files...]
```

Option	Description
<code>-cfgd</code>	Configure la vue à partir de la date et de l'heure spécifiées. Voici des exemples : "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
<code>-cfgl</code>	Configure la vue en utilisant l'étiquette spécifiée. Sans <code>-cfgl</code> , ni <code>-cfgp</code> , ni <code>-cfgd</code> , la configuration actuelle de la vue est utilisée.
<code>-cfgp</code>	Configure la vue en utilisant l'état de promotion spécifié.
<code>-eol</code>	Peut convertir automatiquement les marques de fin de ligne.  Lorsque cette option est activée, les fichiers texte sont transférés depuis le référentiel du serveur vers le dossier de travail de votre station de travail en suivant la convention de fin de ligne instaurée sur la plate-forme exécutant la commande telle que déterminée par la VM Java.  Lorsque cette option est désactivée (par défaut), aucune conversion de fin de ligne n'est exécutée. Utiliser <code>off</code> équivaut à ne pas utiliser <code>-eol</code> du tout.  Lorsque vous spécifiez le caractère de fin de ligne ( <code>cr</code> , <code>lf</code> ou <code>crlf</code> ), les fichiers texte sont transférés depuis le référentiel du serveur vers le dossier de travail de votre station de travail en suivant la convention de fin de ligne spécifiée.  Pour les clients Windows, la marque de fin de ligne est une combinaison retour chariot/saut de ligne ( <code>crlf</code> ) ; pour les plates-formes UNIX, c'est un saut de ligne ( <code>lf</code> ) ; pour les systèmes MAC, un retour chariot ( <code>cr</code> ).  Par exemple, vous définirez cette option par <code>on</code> ou par <code>lf</code> pour comparer un fichier du référentiel avec un fichier de travail présent sur un système UNIX (si le référentiel stocke les fichiers texte avec <code>crlf</code> ).
<code>-f NCO</code>	Spécifie l'extraction de tous les fichiers ayant l'état Manquant ou Obsolète. L'abréviation NCO signifie "needs check-out", c'est-à-dire "à extraire". Aucun autre fichier n'est sélectionné pour l'extraction.  <code>-f NCO</code> est ignoré si <code>-filter</code> est utilisé.
<code>-filter</code>	Spécifie une chaîne d'un ou de plusieurs caractères, chacun d'entre eux représentant un état de fichier. N'incluez jamais d'espace ni aucune autre forme d'espace vierge dans cette chaîne. Seuls les fichiers ayant les états

spécifiés seront extraits. Vous ne pouvez pas extraire les fichiers dont l'état est Pas dans la vue. Les lettres utilisées pour représenter les états sont les suivantes : C = (Current) A jour, M = Modifié, G = (merGe) Fusion, O = Obsolète, I = (mlssing) Manquant et U = (Unknown) Inconnu.

`-filter` a priorité sur `-f NCO`. Si vous utilisez G, M, O ou U, vous devez aussi spécifier `-merge` ou `-o` pour forcer l'opération d'extraction. Si vous ne le faites pas, l'option G, M, O ou U sera ignorée.

---

`-fs` Empêche la mémorisation des états des fichiers après la procédure d'extraction. Par la suite, les valeurs d'état de ces fichiers seront incorrectes et imprécises. Utilisez cette option dans le cas où l'état d'un fichier n'est pas pertinent. Par exemple, si vous supprimez régulièrement les dossiers de travail avant d'extraire les fichiers pour créer un build, il n'y a pas de fichiers et donc leur état est sans importance.

Sachez que les états des fichiers peuvent ne jamais être connus — même si vous utilisez la commande `update-status` ultérieurement. Vous pouvez effectuer une extraction forcée sans l'option `-fs` pour que les fichiers en cours acquièrent l'état correct.

---

`-hook` Cette option ne peut être utilisée qu'avec `-merge`. Elle vous permet de spécifier une autre application (que celle disponible avec l'application) pour effectuer la fusion.

La valeur de cette option doit être le nom du programme à exécuter pour effectuer la fusion, par exemple : `-hook mymerge.sh`

L'application de fusion doit renvoyer un code de sortie : 0 pour montrer qu'aucun conflit n'a été détecté et 1 pour indiquer que des conflits ont été détectés. Toute autre valeur indique une erreur. L'application de fusion ne peut pas être un fichier batch sur les plates-formes Windows car, lorsqu'il est exécuté via Java, le code de sortie n'est pas renvoyé correctement.

Pour chaque fichier à fusionner, `stcmd` fournit trois arguments à l'application de fusion. Ces arguments sont les chemins complets des trois fichiers suivants (dans cet ordre) :

`localFile` : Le fichier de travail local

`commonFile` : Le contenu de la révision de fichier dans le référentiel qui représente la version commune la plus récente entre le fichier en cours de modification localement et la révision spécifiée dans le référentiel

`otherFile` : Le contenu de la révision de fichier en cours d'extraction

Le programme de fusion doit envoyer le résultat de la fusion à la sortie standard (`stdout`).

---

`-i` Permet l'extraction interactive de fichiers dont l'état n'aurait normalement pas permis l'extraction. Le système vous demande confirmation de l'extraction pour chaque fichier ayant l'état Modifié, Fusion ou Inconnu. Votre réponse peut forcer l'extraction du fichier. Si vous utilisez l'option `-i`, vous ne pouvez pas utiliser `-o`, ni `-merge`.

---

`-l` Verrouille chaque fichier en mode non exclusif après son extraction. Si vous n'utilisez ni `-l`, ni `-u`, ni `-nel`, l'état de verrouillage des fichier reste inchangé.

---

`-merge` Vous permet de fusionner le fichier de travail avec la révision en cours d'extraction, et avec la révision sur laquelle ils se basent. Le fichier de travail doit avoir l'état Fusion.

Vous pouvez spécifier l'une des options de création de rapports suivantes avec `-merge` :

`-dryrun` : Indique si le fichier de résultat fusionné contient des conflits ; le fichier de travail local reste inchangé car rien n'est extrait. Cette option fournit une prévisualisation.

`-alwaysprompt` : Demande à l'utilisateur d'enregistrer le fichier de résultat fusionné dans le dossier de travail, qu'il y ait ou non des conflits de fusion.

`-neverprompt` : Enregistre toujours le fichier de résultat fusionné dans le dossier de travail.

`-conflictprompt` : Demande à l'utilisateur d'enregistrer le fichier de résultat fusionné, uniquement si des conflits ont été détectés.

Les options `-dryrun`, `-alwaysprompt`, `-neverprompt` et `-conflictprompt` sont mutuellement exclusives. Si vous n'en spécifiez aucune, le comportement par défaut est `-conflictprompt`.

Vous pouvez spécifier une autre application (que celle disponible avec l'application) pour effectuer la fusion.

Si vous utilisez l'option `-merge`, vous ne pouvez pas utiliser `-i`, ni `-o`.

Si vous n'utilisez pas `-hook` et enregistrez un fichier fusionné comportant des conflits, chaque conflit est marqué de la façon suivante dans le fichier de travail :

```
<<<<<< nomFichier (local)
```

```
ligne telle qu'elle apparaît dans le fichier local
```

```
=====
```

```
ligne telle qu'elle apparaît dans la révision en cours d'extraction
```

```
>>>>>> nomFichier (version x) où nomFichier représente le nom du fichier et x le numéro de la révision en cours d'extraction.
```

Si vous ne spécifiez pas `-merge`, les fichiers ayant l'état Fusion sont ignorés, sauf si vous spécifiez `-o` et forcez l'opération d'extraction.

---

<code>-nel</code>	Verrouille le fichier en mode non exclusif après son extraction.
<code>-o</code>	Force l'extraction des fichiers, quel que soit leur état. Si vous utilisez l'option <code>-o</code> , vous ne pouvez pas utiliser <code>-i</code> , ni <code>-merge</code> .
<code>-ro</code>	Rend le fichier de travail accessible uniquement en lecture après cette opération. Sans cette option, le fichier reste tel qu'il était avant l'opération. Habituellement, vous utiliserez <code>-ro</code> pour vous éviter de modifier un fichier non verrouillé par vous.  L'option <code>-ro</code> doit être utilisée avec <code>-l</code> , ou <code>-u</code> , ou <code>-nel</code> . Si vous utilisez <code>-ro</code> , vous ne pouvez pas utiliser <code>-rw</code> .
<code>-rw</code>	Rend le fichier de travail accessible en lecture et en écriture après cette opération. Sans cette option, le fichier reste tel qu'il était avant l'opération.  L'option <code>-rw</code> doit être utilisée avec <code>-l</code> , ou <code>-u</code> , ou <code>-nel</code> . Si vous utilisez <code>-rw</code> , vous ne pouvez pas utiliser <code>-ro</code> .
<code>-ts</code>	Définit l'horodatage de chaque fichier de travail par l'heure de l'extraction. Sans cette option, le fichier reçoit le même horodatage que sa révision archivée.
<code>-u</code>	Déverrouille les fichiers extraits.
<code>-vd</code>	Spécifie la date/heure utilisée pour identifier les révisions à extraire. C'est la dernière révision avant la date/heure spécifiée qui est extraite pour chaque fichier. Voir plus haut les exemples de date/heure pour <code>-cfgd</code> .
<code>-vl</code>	Spécifie l'étiquette de révision ou de vue utilisée pour identifier les révisions à extraire. Sans l'option <code>-vn</code> , ou <code>-vd</code> , ou <code>-vl</code> , c'est la dernière révision de chaque fichier qui est extraite.
<code>-vn</code>	Spécifie le numéro de révision des fichiers à extraire.

---

## Exemples

L'exemple suivant utilise `stcmd co` pour verrouiller et extraire les fichiers `.doc` situés dans User Manual, enfant du dossier racine StarDraw (dans la vue StarDraw du projet StarDraw).

```
stcmd co -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/User Manual" -l "*.doc"
```

L'exemple suivant utilise `stcmd co` pour fusionner un fichier `readme`.

```
stcmd co -p "NTesla:@10.50.5.179:49201/WebDev/WebDev" -encrypt RC4 -fp  
"/export/home0/johnson/working" -merge "README"
```

## Référence associée

[Opérations en ligne de commande du client](#)

## Comparer les révisions d'un fichier : stcmd diff

Utilisez `stcmd diff` pour afficher les différences entre deux révisions d'un fichier. La commande peut être appliquée à plusieurs fichiers. Si vous ne spécifiez pas de révision avec une option `-vn`, `-vd` ou `-vl`, la copie de travail de chaque fichier spécifié est comparée à la dernière révision présente dans le référentiel (ou espace de stockage) pour ce fichier. Si vous spécifiez une seule révision, la copie de travail de chaque fichier spécifié est comparée à cette révision. Si vous spécifiez deux révisions, les deux révisions de chaque fichier spécifié sont comparées.

Quand vous comparez des fichiers texte, les différences peuvent être affichées. Lors de comparaison de fichiers binaires, les résultats en sortie indiquent si les révisions du fichier sont identiques ou différentes.

### Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd diff -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf]
[-encrypt typeChiffrement] [-cflg "nomEtiquette" | -cflgp "nomEtat" | -cflgd "depuisDate"]
[-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp "cheminDossier"] [-filter "étatFichier"]
[-eol [on | off | cr | lf | crlf]] [-w | -Bpvcs | -b] [-i] [-m "ensembleMasques"] [-t numéro]
[-c numéro] [-n]
[-nd] [-e] [-vl "nomEtiquette" | -vd "depuisDate" | -vn numéroRévision] [fichiers...]
```

Option	Description
<code>-b</code>	Lors de la comparaison de deux lignes de fichiers texte, ignore les espaces de queue (à droite) et traite toutes les autres chaînes d'espaces comme étant de même longueur. Par exemple, les lignes suivantes sont équivalentes : "hi mom" "hi mom"
<code>-Bpvcs</code>	Lors de la comparaison de deux lignes de fichiers texte, ignore les espaces de tête (à gauche) et de queue. Par exemple, les lignes suivantes sont équivalentes car il n'y a qu'un espace entre "hi" et "mom" : "hi mom" "hi mom" mais la ligne suivante n'est pas équivalente : "hi mom"
<code>-c</code>	Spécifie le nombre de ligne inchangées à afficher avant et après la détection d'une différence dans les fichiers texte. Sans cette option, toutes les lignes des fichiers sont affichées. Par exemple, <code>-c 2</code> place deux lignes inchangées avant et après chaque ligne ou ensemble de lignes ayant changé.
<code>-cflgd</code>	Configure la vue à partir de la date et de l'heure spécifiées. Voici des exemples : "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
<code>-cflgl</code>	Configure la vue en utilisant l'étiquette spécifiée. Sans <code>-cflgl</code> , ni <code>-cflgp</code> , ni <code>-cflgd</code> , la configuration actuelle de la vue est utilisée.
<code>-cflgp</code>	Configure la vue en utilisant l'état de promotion spécifié.
<code>-e</code>	Force la commande à renvoyer les codes de sortie suivants : 0 si tous les fichiers comparés sont équivalents 1 en cas d'erreur



2 si au moins un fichier est différent

Vous pouvez utiliser `-e` avec des fichiers texte ou avec des fichiers binaires.

---

<code>-eol</code>	A ce stade, cette commande n'est pas pertinente car <code>diff</code> ignore actuellement les marques de fin de ligne. Si c'est la seule différence entre deux lignes, elles sont indiquées comme étant identiques.
<code>-filter</code>	<p>Spécifie une chaîne d'un ou de plusieurs caractères, chacun d'entre eux représentant un état de fichier. N'incluez jamais d'espace ni aucune autre forme d'espace vierge dans cette chaîne. Seuls les fichiers ayant actuellement les états spécifiés seront comparés. Vous ne pouvez pas comparer les révisions des fichiers dont l'état est Pas dans la vue.</p> <p>Les lettres utilisées pour représenter les états sont les suivantes : <b>C</b> = (Current) A jour, <b>M</b> = Modifié, <b>G</b> = (merGe) Fusion, <b>O</b> = Obsolète, <b>I</b> = (mIssing) Manquant et <b>U</b> = (Unknown) Inconnu.</p>
<code>-i</code>	Ignore la casse des lettres lors de la comparaison de deux fichiers texte. Par exemple, "A" est équivalent à "a".
<code>-m</code>	<p>Quand vous comparez deux fichiers texte, ignore les caractères situés à certaines colonnes que spécifient un ou plusieurs masques. Chaque masque utilise la syntaxe suivante :</p> <pre>"numéroColonne-numéroColonne[ (numeric) ]"</pre> <p>Par exemple, "1-6" ignore les caractères des six premières colonnes de chaque ligne, et "1-6(numeric)" ignore les six premières colonnes de chaque ligne si le caractère dans la colonne 1 est un chiffre dans les deux fichiers.</p> <p>Vous pouvez utiliser une série de masques, mais vous devez les séparer avec des virgules. La syntaxe est :</p> <pre>"masque [,masque] . . ."</pre>
<code>-n</code>	Supprime l'affichage des numéros de ligne dans les deux fichiers texte.
<code>-nd</code>	Supprime l'affichage des différences dans les deux fichiers texte. Quand vous comparez des fichiers binaires, les différences ne sont pas affichées. Cette option est utile avec l'option <code>-e</code> .
<code>-t</code>	Spécifie le nombre d'espaces à utiliser pour chaque tabulation pour l'affichage des différences des fichiers texte. La valeur par défaut est quatre. Utilisez <code>-t 0</code> pour supprimer la conversion des tabulations.
<code>-vd</code>	Spécifie la date/heure utilisée pour identifier les révisions à comparer. C'est la dernière révision avant la date/heure spécifiée qui est utilisée. Voir plus haut les exemples de date/heure pour <code>-cfgd</code> .
<code>-vl</code>	<p>Spécifie l'étiquette de révision ou de vue utilisée pour identifier les révisions à comparer. Vous pouvez spécifier toute combinaison de zéro, une ou deux des options <code>-vn</code>, <code>-vd</code> et <code>-vl</code>.</p> <p>N'utilisez aucune option pour comparer le fichier de travail à la dernière révision, une option pour comparer le fichier de travail à la révision spécifiée et deux options pour comparer deux révisions.</p>
<code>-vn</code>	Spécifie le numéro des révisions à comparer.
<code>-w</code>	<p>Ignore tous les espaces (tabulations et espaces) lors de la comparaison de deux lignes de fichiers texte. Par exemple, les lignes suivantes sont équivalentes :</p> <pre>" a = ( b + 2 );"</pre> <pre>"a=(b+2) ;"</pre> <p>Les options <code>-w</code>, <code>-Bpvcs</code> et <code>-b</code> sont mutuellement exclusives.</p>

---

## Exemple

L'exemple suivant utilise `stcmd diff` pour comparer les révisions Beta1 et Beta2 de chacun des fichiers `.cpp` situés dans le dossier `SourceCode`, enfant du dossier racine `StarDraw` (dans la vue `StarDraw` du projet `StarDraw`). Il ignore tous les espaces.

```
stcmd diff -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/SourceCode" -w -vl  
"Beta1" -vl "Beta2" "*.cpp"
```

### **Référence associée**

[Opérations en ligne de commande du client](#)

## Modifier les descriptions de fichiers : stcmd dsc

Utilisez `stcmd dsc` pour modifier une description de fichier depuis la ligne de commande. Cette commande crée une nouvelle révision de fichier ayant une nouvelle description en guise d'une de ses propriétés.

### Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd dsc -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf] [-encrypt  
typeChiffrement]  
[-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp "cheminDossier"] [-filter "étatFichier"]  
-d "description" [fichiers...]
```

Option	Description
<code>-d</code>	Fournit une description pour les fichiers. Cette description est entourée de guillemets.
<code>-filter</code>	Spécifie une chaîne d'un ou de plusieurs caractères, chacun d'entre eux représentant un état de fichier. N'incluez jamais d'espace ni aucune autre forme d'espace vierge dans cette chaîne. Seuls les fichiers ayant actuellement les états spécifiés recevront la description. Vous ne pouvez pas modifier les descriptions des fichiers dont l'état est Pas dans la vue. Les lettres utilisées pour représenter les états sont les suivantes : C = (Current) A jour, M = Modifié, G = (merGe) Fusion, O = Obsolète, I = (mlssing) Manquant et U = (Unknown) Inconnu.

### Exemple

L'exemple suivant utilise `stcmd dsc` pour modifier la description de `stdafx.cpp` dans `SourceCode`, enfant du dossier racine `StarDraw` (dans la vue `StarDraw` du projet `StarDraw`).

```
stcmd dsc -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/SourceCode" -d  
"SourceCode pour StarTeam" "stdafx.cpp"
```

### Référence associée

[Opérations en ligne de commande du client](#)

## Créer des étiquettes : stcmd label

Utilisez `stcmd label` pour créer une étiquette de vue ou de révision. Une étiquette de vue peut être désignée comme étiquette de build. Par défaut, les étiquettes de vue sont automatiquement appliquées à chaque dossier, fichier, demande de modification, exigence, rubrique et tâche dans la vue. Par défaut, les étiquettes de révision ne sont appliquées à aucun élément.

Vous pouvez utiliser `stcmd apply-label` pour appliquer à des fichiers spécifiques des étiquettes créées à l'aide de `stcmd label`. En outre, vous pouvez utiliser l'option d'étiquette (`-vl`) dans `stcmd ci` pour attacher votre nouvelle étiquette aux fichiers lorsque vous les archivez.

Option	Description
<code>-b</code>	Indique que la nouvelle étiquette est une étiquette de build. Sans <code>-b</code> ou <code>-r</code> , l'étiquette est une étiquette de vue. Les étiquettes de vue (une étiquette de build est un type spécial d'étiquette de vue) sont immédiatement et automatiquement appliquées à chaque dossier, fichier, demande de modification, rubrique et tâche dans la vue.
<code>-d</code>	Spécifie la description de l'étiquette.
<code>-f</code>	Crée la nouvelle étiquette sous forme d'étiquette figée.
<code>-nl</code>	Spécifie le nom de la nouvelle étiquette.
<code>-r</code>	Indique que la nouvelle étiquette est une étiquette de révision. Vous pouvez utiliser la nouvelle étiquette pour les fichiers que vous extrayez. Cette commande n'attache pas la nouvelle étiquette aux éléments, sauf si vous créez cette étiquette en copiant l'étiquette de révision existante attachée à un ou plusieurs éléments. Voir l'option <code>-vl</code> plus bas.
<code>-vd</code>	Spécifie la date/heure pour l'étiquette lorsque vous créez une étiquette de vue. <code>-vd</code> est ignorée si vous créez une étiquette de révision. Sans l'option <code>-vn</code> , ou <code>-vd</code> , ou <code>-vl</code> , l'heure actuelle est utilisée pour les étiquettes de vue. Voici des exemples :  "29/12/01 12:41"  "29 décembre 2001 12:41"  "Monday, December 29, 2001 12:41"
<code>-vl</code>	Spécifie le nom de l'étiquette à copier. Ce nom est entouré de guillemets. L'étiquette spécifiée doit déjà exister dans l'application et doit être du même type que l'étiquette que vous créez. Si l'étiquette spécifiée est une étiquette de révision attachée à un ou plusieurs éléments, la nouvelle étiquette sera attachée à ces mêmes éléments.  Sans l'option <code>-vn</code> , ou <code>-vd</code> , ou <code>-vl</code> , l'heure actuelle est utilisée pour les étiquettes de vue.  S'il s'agit d'une étiquette de révision, vérifiez que vous avez défini <code>-r</code> .
<code>-vp</code>	Spécifie l'état de promotion dont l'étiquette sera copiée. Ce nom doit être entouré de guillemets. L'étiquette spécifiée doit déjà exister dans l'application, et vous devez créer une étiquette de vue pour que cette option soit correcte.  Sans l'option <code>-vn</code> , ou <code>-vd</code> , ou <code>-vl</code> , l'heure actuelle est utilisée pour les étiquettes de vue.  Vous pouvez créer une étiquette de vue basée sur un état de promotion UNIQUEMENT quand l'état de promotion est défini par une étiquette et non par <a href="#">A jour</a> .

## Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd label -p "nomUtilisateur:motPasse@hôte:port/projet/vue/dossier" [-pwdfile  
"cheminFichier"] [-cmp] [-encrypt typeChiffrement]  
[-q] [-x] [-stop] -nl "nomEtiquette" [-vl "nomEtiquette" | -vd "depuisDate" | -vp nomEtat]  
[-d "description"] [-b | -r] [-f]
```

## Exemple

L'exemple suivant utilise stcmd label pour créer une nouvelle étiquette de build nommée Beta pour la vue StarDraw du projet StarDraw.

```
stcmd label -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw" -nl "Beta" -b
```

## Référence associée

[Opérations en ligne de commande du client](#)

## Créer des dossiers de travail : stcmd local-mkdir

Utilisez `stcmd local-mkdir` afin de créer sur votre station de travail le dossier de travail ou le répertoire de travail pour le dossier StarTeam spécifié. En outre, utilisez l'option `-is` pour créer des dossiers de travail (ou répertoires de travail) pour les dossiers enfant du dossier StarTeam spécifié.

### Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd local-mkdir -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf]
[-encrypt typeChiffrement] [-cagl "nomEtiquette"] -cagp "nomEtat" | -cagd "depuisDate"]
[-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp "cheminDossier"]
```

Option	Description
<code>-cagd</code>	Configure la vue à partir de la date et de l'heure spécifiées. Voici des exemples : "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
<code>-cagl</code>	Configure la vue en utilisant l'étiquette spécifiée. Sans <code>-cagl</code> , ni <code>-cagp</code> , ni <code>-cagd</code> , la configuration actuelle de la vue est utilisée.
<code>-cagp</code>	Configure la vue en utilisant l'état de promotion spécifié.

### Exemple

L'exemple suivant utilise `stcmd local-mkdir` pour créer les dossiers de travail de SourceCode, enfant du dossier racine StarDraw (dans la vue StarDraw du projet StarDraw) et ses dossiers enfant.

```
stcmd local-mkdir -p "JMarsh:motPasse@Orion:1024/ StarDraw/StarDraw/SourceCode" -is
```

### Référence associée

[Opérations en ligne de commande du client](#)

## Supprimer des fichiers locaux : stcmd delete-local

Utilisez `stcmd delete-local` pour supprimer des fichiers d'un dossier de travail. Vous pouvez supprimer les fichiers placés sous contrôle de version, ainsi que ceux ne figurant pas dans StarTeam. Cette action ne retire pas les fichiers du contrôle de version. Elle réduit simplement la quantité des données stockées sur votre station de travail dans vos dossiers de travail. Si vous supprimez des fichiers en fonction de leur état StarTeam, il est conseillé d'utiliser d'abord `stcmd update-status`.

### Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd delete-local -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf]
[-encrypt typeChiffrement] [-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp
"cheminDossier"]
[-cagl "nomEtiquette" | -cagp "nomEtat" | -cagd "depuisDate"] [-filter "étatFichier"]
[fichiers...]
```

Option	Description
<code>-cagd</code>	Configure la vue à partir de la date et de l'heure spécifiées. Voici des exemples : "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
<code>-cagl</code>	Configure la vue en utilisant l'étiquette spécifiée. Sans <code>-cagl</code> , ni <code>-cagp</code> , ni <code>-cagd</code> , la configuration actuelle de la vue est utilisée.
<code>-cagp</code>	Configure la vue en utilisant l'état de promotion spécifié.
<code>-filter</code>	Spécifie une chaîne d'un ou de plusieurs caractères, chacun d'entre eux représentant un état de fichier. N'incluez jamais d'espace ni aucune autre forme d'espace vierge dans cette chaîne. Seuls les fichiers ayant actuellement les états spécifiés seront supprimés. Les lettres utilisées pour représenter les états sont les suivantes : C = (Current) A jour, M = Modifié, G = (merGe) Fusion, O = Obsolète, I = (missing) Manquant et U = (Unknown) Inconnu.

### Exemple

L'exemple suivant utilise `stcmd delete-local` pour supprimer certains fichiers du dossier de travail du dossier StarTeam nommé `SourceCode`. `SourceCode` est un enfant du dossier racine `StarDraw` (dans la vue `StarDraw` du projet `StarDraw`). Cet exemple supprime tous les fichiers non placés sous contrôle de version. Ces fichiers ont pour état **Pas dans la vue**.

```
stcmd delete-local -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/SourceCode"
-filter "N" "*"
```

### Référence associée

[Opérations en ligne de commande du client](#)

# Afficher l'historique des fichiers : stcmd hist

Utilisez `stcmd hist` pour afficher l'historique des révisions des fichiers.

## Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd hist -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf]
[-encrypt typeChiffrement] [-cflg "nomEtiquette" | -cflg "nomEtat" | -cflg "depuisDate"]
[-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp "cheminDossier"] [-filter "étatFichier"]
[fichiers...]
```

Option	Description
<code>-cflg</code>	Configure la vue à partir de la date et de l'heure spécifiées. Voici des exemples : "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
<code>-cflg</code>	Configure la vue en utilisant l'étiquette spécifiée. Sans <code>-cflg</code> , ni <code>-cflg</code> , ni <code>-cflg</code> , la configuration actuelle de la vue est utilisée.
<code>-cflg</code>	Configure la vue en utilisant l'état de promotion spécifié.
<code>-filter</code>	Spécifie une chaîne d'un ou de plusieurs caractères, chacun d'entre eux représentant un état de fichier. N'incluez jamais d'espace ni aucune autre forme d'espace vierge dans cette chaîne. Seuls les fichiers ayant actuellement les états spécifiés seront inclus dans le rapport. Vous ne pouvez pas afficher l'historique des fichiers dont l'état est Pas dans la vue.  Les lettres utilisées pour représenter les états sont les suivantes : C = (Current) A jour, M = Modifié, G = (merGe) Fusion, O = Obsolète, I = (mlssing) Manquant et U = (Unknown) Inconnu.

## Exemple

L'exemple suivant utilise `stcmd hist` pour afficher l'historique des révisions du fichier `star.h` situé dans `SourceCode`, enfant du dossier racine `StarDraw` (dans la vue `StarDraw` du projet `StarDraw`).

```
stcmd hist -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/SourceCode" "star.h"
```

## Référence associée

[Opérations en ligne de commande du client](#)



## Lister des fichiers : stcmd list

Utilisez `stcmd list` pour lister tous les fichiers situés dans le dossier spécifié par l'option `-p`. Les fichiers sont ceux qui existaient à un moment donné ou qui possèdent une étiquette spécifique.

### Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd list -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf]
[-encrypt typeChiffrement] [-cflg "nomEtiquette" | -cfgp "nomEtat" | -cfgd "depuisDate"]
[-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp "cheminDossier"] [-filter "étatFichier"]
[-cf] [fichiers...]
```

Option	Description
<code>-cf</code>	Force l'ajout à la liste des noms de dossiers enfant présents dans le dossier.
<code>-cfgd</code>	Configure la vue à partir de la date et de l'heure spécifiées. Voici des exemples : "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
<code>-cflg</code>	Configure la vue en utilisant l'étiquette spécifiée. Sans <code>-cflg</code> , ni <code>-cfgp</code> , ni <code>-cfgd</code> , la configuration actuelle de la vue est utilisée. <code>-cfgp</code> configure la vue en utilisant l'état de promotion spécifié.
<code>-filter</code>	Spécifie une chaîne d'un ou de plusieurs caractères, chacun d'entre eux représentant un état de fichier. N'incluez jamais d'espace ni aucune autre forme d'espace vierge dans cette chaîne. Seuls les fichiers ayant actuellement les états spécifiés seront inclus dans la liste. Vous ne pouvez pas inclure les fichiers dont l'état est Pas dans la vue.  Les lettres utilisées pour représenter les états sont les suivantes : C = (Current) A jour, M = Modifié, G = (merGe) Fusion, O = Obsolète, I = (mlssing) Manquant et U = (Unknown) Inconnu.
<code>-short</code>	Fournit une liste courte simplifiée des fichiers locaux et de leur état avec l'abréviation de l'état et le chemin relatif du fichier de travail, par exemple : M /starteam/Server.java N /starteam/LabelInfo.java  Sans cette option, la liste se compose ainsi :  Une ligne pour chaque nom de dossier suivi du chemin de son dossier de travail. Dans un dossier, une ligne pour chaque fichier commençant par l'état non abrégé et contenant les droits, l'horodatage et le nom du fichier.  Par exemple :  Dossier : Source (rép. de travail : E:\Source) Inconnu rw 6/4/02 19:42:18 230 req.bmp

### Exemple

L'exemple suivant utilise `stcmd list` pour lister tous les fichiers situés dans `SourceCode`, enfant du dossier racine `StarDraw` (dans la vue `StarDraw` du projet `StarDraw`), ainsi que tous les fichiers stockés dans les dossiers enfant de `SourceCode`.

```
stcmd list -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/SourceCode" -is "*"
```

`-short` fournit une liste courte simplifiée des fichiers locaux et de leur état avec l'abréviation de l'état et le chemin relatif du fichier de travail, par exemple :

```
M /starteam/Server.java N /starteam/LabelInfo.java.
```

Sans l'option `-short`, la liste consiste en une ligne pour chaque nom de dossier suivi du chemin de son dossier de travail. Dans un dossier, la liste contient une ligne pour chaque fichier commençant par l'état non abrégé et contenant les droits, l'horodatage et le nom du fichier. Par exemple :

```
Dossier : Source (rép. de travail : E:\Source) Inconnu rw 6/4/02 19:42:18 230 req.bmp
```

## Référence associée

[Opérations en ligne de commande du client](#)

# Verrouiller et déverrouiller des fichiers : stcmd lck

Utilisez `stcmd lck` pour verrouiller ou déverrouiller des fichiers depuis la ligne de commande.

Option	Description
<code>-break</code>	Suspend le verrou d'un autre utilisateur si vous avez les droits d'accès appropriés.
<code>-filter</code>	Spécifie une chaîne d'un ou de plusieurs caractères, chacun d'entre eux représentant un état de fichier. N'incluez jamais d'espace ni aucune autre forme d'espace vierge dans cette chaîne. Seuls les fichiers ayant actuellement les états spécifiés seront verrouillés ou déverrouillés. Vous ne pouvez pas verrouiller ni déverrouiller les fichiers dont l'état est Pas dans la vue. Les lettres utilisées pour représenter les états sont les suivantes : C = (Current) A jour, M = Modifié, G = (merGe) Fusion, O = Obsolète, I = (mIssing) Manquant et U = (Unknown) Inconnu.
<code>-l</code>	Verrouille les fichiers. C'est l'option par défaut si vous n'utilisez ni <code>-l</code> , ni <code>-nel</code> , ni <code>-u</code> .
<code>-nel</code>	Verrouille les fichiers en mode non exclusif.
<code>-ro</code>	Rend le fichier de travail accessible uniquement en lecture après cette opération. Sans cette option, le fichier reste tel qu'il était avant l'opération. Habituellement, vous utilisez <code>-ro</code> pour vous empêcher de modifier un fichier non verrouillé par vous.  L'option <code>-ro</code> doit être utilisée avec <code>-l</code> , <code>-u</code> , ou <code>-nel</code> . Si vous utilisez <code>-ro</code> , vous ne pouvez pas utiliser <code>-rw</code> .
<code>-rw</code>	Rend le fichier de travail accessible en lecture et en écriture après cette opération. Sans cette option, le fichier reste tel qu'il était avant l'opération.  L'option <code>-rw</code> doit être utilisée avec <code>-l</code> , <code>-u</code> , ou <code>-nel</code> . Si vous utilisez <code>-rw</code> , vous ne pouvez pas utiliser <code>-ro</code> .
<code>-u</code>	Déverrouille les fichiers.

## Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd lck -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf]
[-encrypt typeChiffrement] [-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp
"cheminDossier"]
[-filter "étatFichier"] [-break] [-l | -u | -nel] [-ro | -rw]] [fichiers...]
```

## Exemple

L'exemple suivant utilise `stcmd lck` pour déverrouiller tous les fichiers situés dans SourceCode, enfant du dossier racine StarDraw (dans la vue StarDraw du projet StarDraw), ainsi que tous les fichiers stockés dans les dossiers enfant de SourceCode.

```
stcmd lck -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/SourceCode" -is -u "*" 
```

## Référence associée

[Opérations en ligne de commande du client](#)

# Verrouiller et déverrouiller un serveur : stcmd server-mode

Si vous disposez des droits d'accès ou des privilèges appropriés, vous pouvez utiliser `stcmd server-mode` pour verrouiller et déverrouiller une configuration de serveur. Le verrouillage d'une configuration de serveur restreint l'accès à celle-ci lorsque vous réalisez une sauvegarde ou exécutez d'autres procédures sensibles. Lorsque le serveur est verrouillé, seules les commandes d'administration du serveur sont acceptées. Lorsque la configuration de serveur est déverrouillée, les opérations ordinaires reprennent.

**Remarque :** Si le nom d'utilisateur est "StarTeam", cette commande requiert un mot de passe, même si un mot de passe a déjà été fourni ou si l'utilisateur a un mot de passe vierge.

## Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd server-mode [-pwdfile "cheminFichier"] [-cmp] [-encrypt typeChiffrement] [-q] [-x] [-stop] -s "nomServeur" -mode [lock | exlock | unlock]
```

Option	Description
<code>exlock</code>	Verrouille le serveur en mode exclusif pour que personne d'autre ne puisse y accéder.
<code>lock</code>	Verrouille le serveur en mode non exclusif. Vous ne pouvez effectuer que des commandes administratives.
<code>-mode</code>	Indique si le serveur doit être verrouillé, verrouillé en mode exclusif ou déverrouillé.  Si vous utilisez la commande <code>-mode</code> , seules les commandes d'administration du serveur sont acceptées jusqu'au déverrouillage du serveur. Par exemple, vous pouvez l'utiliser quand vous exécutez un programme de sauvegarde.  Si vous utilisez la commande <code>-mode exlock</code> , vous ne pouvez pas accéder au serveur jusqu'à ce qu'il soit déverrouillé. Par exemple, vous pouvez l'utiliser quand vous créez un champ personnalisé.  Utilisez <code>-mode unlock</code> pour rendre le serveur à nouveau disponible aux autres utilisateurs.
<code>unlock</code>	Déverrouille le serveur pour que toute personne disposant des droits d'accès appropriés puisse y accéder.
<code>-s</code>	Spécifie le serveur. La syntaxe complète est : <pre>-s "nomUtilisateur:motPasse@nomHôte:numéroPort"</pre> Par exemple : <pre>-s "JMarsh:motPasse@orion:49201"</pre> Si le nom d'utilisateur est omis, le nom d'utilisateur actuel est utilisé. Le nom d'utilisateur spécifié dans l'exemple est "JMarsh"  Si le mot de passe est omis, l'utilisateur est invité à le saisir. Le mot de passe spécifié dans l'exemple est "motPasse"  Si le nom d'hôte est omis, la valeur par défaut est localhost. Le nom d'hôte spécifié dans l'exemple est "orion". Le numéro de port est requis. L'exemple utilise le numéro de port par défaut : 49201.

## Exemple

L'exemple suivant utilise `stcmd server-mode` pour verrouiller le serveur qui utilise le port 1024 sur Orion.

```
stcmd server-mode -s "JMarsh:motPasse@Orion:1024" -mode lock
```

**Référence associée**

[Opérations en ligne de commande du client](#)

## Supprimer des fichiers : stcmd remove

Utilisez `stcmd remove` pour retirer des fichiers du contrôle de version. Les fichiers spécifiés et l'historique de leurs révisions n'apparaissent plus dans StarTeam sauf si vous faites revenir la vue du projet à un point antérieur à leur retrait.

### Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd remove -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf]
[-encrypt typeChiffrement] [-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp
"cheminDossier"]
[-filter "étatFichier"] [-df] [fichiers...]
```

Option	Description
<code>-df</code>	Supprime le fichier de travail de l'utilisateur. Sans cette option, le fichier de travail reste dans le dossier de travail sur votre station de travail.
<code>-filter</code>	Spécifie une chaîne d'un ou de plusieurs caractères, chacun d'entre eux représentant un état de fichier. N'incluez jamais d'espace ni aucune autre forme d'espace vierge dans cette chaîne. Seuls les fichiers ayant actuellement les états spécifiés seront supprimés. Vous ne pouvez pas supprimer du système de contrôle des versions les fichiers qui ont l'état Pas dans la vue. Les lettres utilisées pour représenter les états sont les suivantes : C = (Current) A jour, M = Modifié, G = (merGe) Fusion, O = Obsolète, I = (mlssing) Manquant et U = (Unknown) Inconnu.

### Exemple

L'exemple suivant utilise `stcmd remove` pour retirer tous les fichiers .hm de SourceCode, enfant du dossier racine StarDraw (dans la vue StarDraw du projet StarDraw), ainsi que tous les fichiers stockés dans les dossiers enfant de SourceCode. En outre, il supprime les fichiers de travail.

```
stcmd remove -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/SourceCode" -is -df "*.hm"
```

### Référence associée

[Opérations en ligne de commande du client](#)

## Définir les options personnelles : stcmd set-personal-options

Utilisez `stcmd set-personal-options` pour définir et lister les options personnelles. A ce jour, les seules options personnelles pouvant être définies avec cette commande sont :

- ◆ Le mode de stockage des informations d'état des fichiers.
- ◆ L'emplacement de stockage des informations d'état des fichiers, lorsque celles-ci sont stockées de façon centralisée.

Les informations d'état des fichiers sont stockées dans le fichier `starteam-client-options.xml`. Si aucun fichier `starteam-client-options.xml` n'existe, cette commande le crée dans l'emplacement par défaut. L'emplacement par défaut est la valeur par défaut de l'option `central-status` (voir ci-après).

### Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd set-personal-options [-q] [-x] [-stop] [-central-status | -per-folder-status]
[-central-repository "cheminDossier"] [-list]
```

Option	Description
<code>-central-repository</code>	Vous permet de spécifier un emplacement pour le référentiel central.
<code>-central-status</code>	<p>Indique que les informations d'état des fichiers seront stockées à un emplacement central pour cet utilisateur sur cette station de travail. Vous pouvez définir cet emplacement avec le client Windows ou le client multiplate-forme, ou avec cette commande (voir l'option <code>-central-repository</code>).</p> <p>Si vous ne définissez pas cet emplacement, il est défini par défaut comme suit :</p> <p>Pour Windows NT, le dossier dans lequel l'application a été installée</p> <p>Pour Windows XP : <code>C:\Documents and Settings\nomutilisateur\Local Settings\Application Data\Borland\StarTeam\</code></p> <p>Pour les plates-formes non Windows : <code>/répertoire_home_utilisateur/.starteam-client/</code></p>
<code>-per-folder-status</code>	Indique que chaque dossier de travail contient des informations sur l'état de ses propres fichiers. Ces informations sont stockées dans le dossier <code>.sbas</code> , qui est un dossier enfant du dossier de travail. Si vous déplacez le dossier de travail, les informations d'état des fichiers se déplacent avec lui.
<code>-list</code>	Affiche la liste des options personnelles actuellement définies dans le fichier <code>starteam-client-options.xml</code> qui les stocke.

### Exemple

L'exemple suivant utilise `stcmd set personal-options` pour indiquer que les informations d'état des fichiers seront stockées de façon centralisée dans `C:\JMarsh\statusinfo`.

```
stcmd set-personal-options -central-status -central-repository "C:\JMarsh\statusinfo"
```

**Référence associée**

[Opérations en ligne de commande du client](#)



## Mettre à jour l'état d'un fichier : stcmd update-status

Lorsque vous mettez à jour l'état d'un fichier, StarTeam compare le fichier de travail à la révision que vous avez extraite et à la dernière révision. Par exemple, votre liste de fichiers peut indiquer que le fichier a pour état A jour, alors que quelqu'un d'autre vient d'en archiver une copie, ce qui lui attribue de fait l'état Obsolète.

La mise à jour de l'état des fichiers n'est pas la mise à jour des fichiers. Si un fichier ne figure pas dans votre dossier de travail, la mise à jour de l'état vous permet de savoir que l'état du fichier est Manquant mais elle n'extrait pas le fichier. Normalement, vous mettez à jour l'état d'un fichier pour déterminer si celui-ci doit être archivé, extrait, ajouté ou ignoré.

Par exemple, vous pouvez :

- ◆ Archiver un fichier ayant l'état Obsolète, Manquant ou Fusion.
- ◆ Extraire un fichier ayant l'état Modifié ou Fusion.
- ◆ Ajouter à l'application un fichier ayant l'état Pas dans la vue. Cependant, la commande `update-status` ne liste jamais les fichiers ayant l'état Pas dans la vue car ils ne sont pas stockés dans le référentiel.

Utilisez `stcmd update-status` pour afficher le nom du fichier, son état avant l'exécution de la commande et son état après l'exécution de celle-ci. Le résultat obtenu peut être la ligne suivante : `x.cpp: l'état est A jour (c'était Inconnu)`.

## Syntaxe

La syntaxe de cette commande est la suivante :

```
stcmd update-status -p "spécificateurProjet" [-pwdfile "cheminFichier"] [-cmp] [-csf]
[-encrypt typeChiffrement] [-is] [-q] [-x] [-stop] [-rp "cheminDossier" | -fp
"cheminDossier"]
[-cagl "nomEtiquette" | -cagp "nomEtat" | -cagd "depuisDate"] [-filter "étatFichier"] [-eol
[on | off ]][-v] [fichiers...]
```

Option	Description
<code>-cagd</code>	Configure la vue à partir de la date et de l'heure spécifiées. Voici des exemples : "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
<code>-cagl</code>	Configure la vue en utilisant l'étiquette spécifiée. Sans <code>-cagl</code> , ni <code>-cagp</code> , ni <code>-cagd</code> , la configuration actuelle de la vue est utilisée.
<code>-cagp</code>	Configure la vue en utilisant l'état de promotion spécifié.
<code>-eol</code>	Quand cette option est activée, calcule l'état du fichier texte sans utiliser les marques de fin de ligne (eol). Quand elle est désactivée (par défaut), l'état de mise à jour est calculé sur le fichier de travail et les dernières révisions avec leurs marques actuelles de fin de ligne. Quand cette option est activée, un fichier de travail ayant des sauts de ligne pour marques de fin de ligne, peut être comparé à une dernière révision ayant des retours chariots/sauts de ligne et considéré A jour si seules les marques de fin de ligne diffèrent.
<code>-contents</code>	Envoie le contenu du fichier au lieu de la somme de contrôle MD5.
<code>-filter</code>	Spécifie une chaîne d'un ou de plusieurs caractères, chacun d'entre eux représentant un état de fichier. N'incluez jamais d'espace ni aucune autre forme d'espace vierge dans cette chaîne. Seuls les fichiers ayant actuellement les états spécifiés seront mis à jour. <b>C</b> = (Current) A jour, <b>M</b> = Modifié, <b>G</b> = (merGe) Fusion, <b>O</b> = Obsolète, <b>I</b> = (missing) Manquant et <b>U</b> = (Unknown) Inconnu.

---

-v

Indique l'état de chaque fichier dans le dossier de travail du dossier, sauf s'il a l'état Pas dans la vue. Sans cette option, l'état d'un fichier n'est affiché que s'il a changé.

---

## Exemple

L'exemple suivant utilise `stcmd update-status` pour vérifier que chaque fichier du dossier de travail du dossier StarTeam nommé `SourceCode` possède un état précis. `SourceCode` est un enfant du dossier racine `StarDraw` (dans la vue StarDraw du projet StarDraw).

```
stcmd update-status -p "JMarsh:motPasse@Orion:1024/StarDraw/StarDraw/SourceCode" "*"
```

## Référence associée

[Opérations en ligne de commande du client](#)

## Utilitaires VCM en ligne de commande

L'utilitaire Comparaison/Fusion de vues en ligne de commande ([VCMUtility](#)) compare une vue StarTeam source à une vue cible et, de manière facultative, fusionne les différences dans la vue cible.

Cette section comprend toutes les rubriques d'aide relatives à [VCMUtility](#).

Comparaison/Fusion de vues est uniquement disponible dans le Client multiplate-forme StarTeam et dans [VCMUtility](#).

### Dans cette section

#### [Présentation de l'utilitaire VCM en ligne de commande \(VCMUtility\)](#)

Explique l'utilitaire Comparaison/Fusion de vues en ligne de commande qui se nomme [VCMUtility](#).

#### [Commandes de VCMUtility](#)

Explique les types des commandes utilisées en ligne par l'utilitaire Comparaison/Fusion de vues ([VCMUtility](#)).

#### [Options de connexion de VCMUtility](#)

Explique les options de connexion utilisées en ligne par l'utilitaire Comparaison/Fusion de vues ([VCMUtility](#)).

#### [Options de session de VCMUtility](#)

Cette section décrit les options de [VCMUtility](#) qui peuvent être utilisées dans les nouvelles sessions.

#### [Options diverses de VCMUtility](#)

Explique les options diverses utilisées en ligne par l'utilitaire Comparaison/Fusion de vues ([VCMUtility](#)).

#### [Exemples VCMUtility](#)

Donne des exemples de l'utilisation de ([VCMUtility](#)).

#### [Aide-mémoire](#)

Liste les commandes et les options de [VCMUtility](#) accompagnées de leur syntaxe.

#### [Syntaxe des options composées de VCMUtility](#)

Explique en détail la syntaxe de chacune des options de [VCMUtility](#).

# Présentation de l'utilitaire VCM en ligne de commande (VCMUtility)

VCMUtility est un utilitaire en ligne de commande qui compare une vue StarTeam source à une vue cible et, de manière facultative, fusionne les différences dans la vue cible.

La documentation VCMUtility inclut ce qui suit :

- ◆ Commandes
- ◆ Options de connexion
- ◆ Options de session
- ◆ Options diverses non sauvegardées avec les sessions

**Remarque :** Vous pouvez commencer une session de Comparaison/Fusion de vues en utilisant la ligne de commande et la finir dans le Client multiplate-forme StarTeam en utilisant l'interface utilisateur de Comparaison/Fusion de vues. Par exemple, vous pouvez utiliser VCMUtility pour créer une session VCM, en utilisant peut-être l'option `DefaultAction` mais sans la laisser aller jusqu'à la validation. Il enregistrera automatiquement la session VCM si nécessaire sous n'importe quel autre nom que vous choisirez. Vous pouvez ensuite ouvrir cette session VCM dans le client StarTeam multiplate-forme, la réviser et faire des ajustements, puis valider les modifications dans le référentiel.

## Conventions syntaxiques

La syntaxe de la ligne de commande utilise les conventions suivantes :

Convention	Description
Accolades { }	Entoure les éléments syntaxiques obligatoires.
Crochets [ ]	Entoure les éléments facultatifs
Crochets angulaires < >	Entoure un mot ou une phrase devant être remplacé par une valeur ou un jeu de valeurs. Par exemple, <nom fichier> sera remplacé par le véritable nom ou chemin du fichier, et <idutilisateur> sera remplacé par un véritable ID d'utilisateur. Cependant, beaucoup de mots ou de phrases entre crochets angulaires peuvent se développer en une syntaxe plus complexe. Par exemple, <demandes de modification> peut être remplacé par CR, CRs, ChangeRequest, ChangeRequests, ChangeRequest *4277, etc. Lorsque vous hésitez, reportez-vous à la rubrique "Aide-mémoire" VCMUtility de la section <a href="#">Reference/CompareMerge</a> de la documentation. L'"Aide-mémoire" décrit la syntaxe complète des phrases comme <demandes de modification>.
Barre verticale	Sépare des éléments qui constituent une alternative.
Astérisque * servant de préfixe	Indique que l'élément qui suit peut être répété.

**Remarque :** Aucune option ne distingue les majuscules des minuscules (par exemple, `Server` est identique à `server`).

## Commande de VCMUtility

```
VCMUtility [<fichier options>] [options]
```

Vous pouvez fournir les options avec le fichier <fichier options> spécifié (comme premier paramètre), avec les arguments de la ligne de commande, ou avec les deux. Les arguments de la ligne de commande remplacent

toutes les options trouvées dans le <fichier options>. Dans le <fichier options>, le nom des options doit commencer à la première position d'une nouvelle ligne et exclure le caractère "-".

## Fichier d'options de VCMUtility

Vous pouvez spécifier les options de `VCMUtility` dans un fichier d'options dont le nom est transmis comme premier paramètre de la commande `VCMUtility`.

Exemple :

```
VCMUtility c:\VCMconfig.txt
```

Dans le fichier, chaque option doit commencer sur une nouvelle ligne. Les noms des options doivent commencer en colonne 1 et être suivis par au moins un caractère espace. La valeur d'une option peut occuper plusieurs lignes si chaque ligne de continuation commence par un espace ou par caractère de tabulation. Les lignes blanches sont ignorées. Vous pouvez fournir des commentaires en les faisant précéder d'une double barre oblique (//).

Exemple :

```
// Ceci est un commentaire
server jsmith:monmpasse@unhôte:49201
type Rebase
include "/Cygnus/StarTeam/<Noyau StarTeam>/Server/Common/*.h" +ALL
*.cpp *.rc Makefile // valeur longue continuée sur une seconde ligne

// La ligne ci-dessus était blanche
save
ma-session-refondation // valeur fournie sur une ligne séparée
```

## Paramètres de la ligne de commande

Les options de `VCMUtility` peuvent être transmises en tant que paramètres de la ligne de commande en faisant précéder leur nom d'un tiret. Par exemple, l'option `Server` peut être fournie en tant que le paramètre de la ligne de commande `-server`. Si une option a des unités de "valeur" secondaires, elles doivent suivre immédiatement le nom de l'option (sans utiliser de tiret).

## Panachage des sources en entrée

Les options de `VCMUtility` peuvent être fournies dans un fichier d'options, avec les paramètres de la ligne de commande ou en panachant les deux sources. Par exemple, les valeurs d'options les plus utilisées, ou valeurs "statiques", peuvent être placées dans le fichier de configuration alors que les valeurs "dynamiques" peuvent être fournies par les paramètres de la ligne de commande.

Un paramètre de la ligne de commande peut spécifier une option déjà définie dans le fichier configuration. Lorsqu'un argument de la ligne de commande spécifie la même option que le fichier d'options, la valeur de l'option présente sur la ligne de commande remplace la valeur de l'option lue dans le fichier de configuration. Par exemple, si le fichier de configuration spécifie `Source Vue1` et si la ligne de commande spécifie `-Source Vue2`, `Vue2` servira de vue source.

## Valeurs des options utilisant les caractères Unicode

L'encodage des valeurs des options transmises comme arguments de la ligne de commande est contrôlé par l'environnement qui a lancé cette dernière (par exemple, le shell de commande). En conséquence, sur les systèmes

où des valeurs d'options devant être transmises à `VCMUtility` utilisent des caractères non exprimables par l'environnement de lancement, ces options doivent être transmises par le biais du fichier d'options.

Lorsque que le fichier d'options ne commence pas par une marque d'ordre des octets (BOM, byte-order mark), il est ouvert avec le jeu de caractères par défaut du système (par exemple, ANSI [Windows-1252] sous Windows, UTF-8 sous Linux). Si le fichier d'options commence par une BOM, il est interprété avec l'encodage correspondant. Les encodages UTF-8 et UTF-16 permettent de fournir le jeu complet des caractères Unicode dans le fichier d'options.

Pour référence, les séquences BOM sont les suivantes :

BOM	Encodage
0xEFBBBF	UTF-8
0xFEFF	UTF-16 BE (big-endian)
0xFFFE	UTF-16 LE (little-endian)

## Options booléennes

La valeur par défaut de toutes les options booléennes (`True` ou `False`) est `False`. Cependant, spécifier une option booléenne sans spécifier sa valeur est équivalent à spécifier la valeur `True`. De ce fait, une option booléenne peut être activée en la citant. Exemple :

```
// Définit ces options par True
AutoLogon
BreakLocks
```

## Abréviations

Outre leur "forme longue" (employée dans ce document), la majorité des noms de commandes et d'options ont une ou plusieurs "formes courtes", ou abréviations. Ces orthographes permettent d'écourter les commandes de `VCMUtility` ayant de nombreuses options. Les listes complètes des abréviations sont fournies par la commande `--Help abbreviations`. Voici des exemples d'abréviations :

`Help` : `H` ou `?`.

`ActiveProcessItem` : `ActivePI` ou `API`.

`SourceLabel` : `SrcLabel` ou `SL`.

Dans la plupart des cas, un élément syntaxique orthographié dans ce document en mêlant majuscules et minuscules, peut être abrégé en conservant uniquement les majuscules. Par exemple, `ManualMergeFiles` peut s'abrégé en `MMF` ou `AutoMergeProperties` peut s'abrégé en `AMP`, etc.

## Codes de sortie

L'utilitaire VCM renvoie les codes de sortie suivants pour indiquer le résultat de son exécution :

Code de sortie	Description
0	Pas d'erreur.
1	Une erreur fatale s'est produite.

## Fichiers journal de VCMUtility

Au cours de son exécution, `VCMUtility` écrit les messages d'avertissement et d'erreur dans la fenêtre console (sortie standard). Pour la plupart des opérations, `VCMUtility` crée aussi un fichier journal qui résume l'opération. Comme pour la sortie dans la fenêtre console, le fichier journal est plus détaillé quand l'option `Verbose` est activée. Le fichier journal est créé pour les nouvelles sessions VCM et pour les commandes `Import`, `Open`, `Replay` et `Resume`. Cependant, le fichier journal n'est pas démarré tant que les paramètres de la ligne de commande et le fichier d'options, s'il est utilisé, n'ont pas été analysés comme exempts d'erreur. Aucun fichier journal n'est créé pour les commandes `Help` ou `Delete`.

Le fichier journal de `VCMUtility` est créé dans le répertoire de base de l'utilisateur (identifié par Java par `user.home`) avec le titre suivant :

```
VCMUtility-AAAA-MM-JJ_hh-mm-ss.log
```

`AAAA-MM-JJ` et `hh-mm-ss` sont la date et l'heure en cours dans le fuseau horaire local. Le nom de chemin complet du fichier journal s'écrit dans la fenêtre console lorsque le fichier journal est démarré.

## Prise en charge des packages de modifications par VCMUtility

A partir de la version StarTeam 2009, `VCMUtility` prend en charge les packages de modifications pour toute configuration StarTeam qui a été mise au niveau de la version 2009. Comme les packages de modifications sont des objets persistants stockés sur le serveur, ils offrent de nombreux avantages par rapport aux fichiers de session VCM (`.vcms`) et d'exportation VCM (`.vcmx`). C'est pourquoi, pour les configurations StarTeam qui ont été mises à niveau, les packages de modifications sont préférés aux fichiers de session et d'exportation pour l'enregistrement et la reprise des sessions. De même, l'option `Save` sans paramètre et la commande `Open` sont préférées à l'option `Save` avec un paramètre et aux commandes `Resume`, `Export` et `Import`. Cependant, pour des raisons de compatibilité arrière, `VCMUtility` 2009 prend encore en charge les commandes utilisant les fichiers de session VCM. Voir la commande `Open` et l'option `Save`, pour de plus amples informations.

### Référence associée

- [Commandes de VCMUtility](#)
- [Options de connexion de VCMUtility](#)
- [Options de session de VCMUtility](#)
- [Options diverses de VCMUtility](#)
- [Exemples VCMUtility](#)
- [Aide-mémoire](#)

# Commandes de VCMUtility

Cette section définit les fonctionnalités de `VCMUtility` par rapport aux commandes exécutant l'utilitaire. Chaque exécution de `VCMUtility` exécute une commande.

Pour la description des conventions syntaxiques, voir "Présentation de l'utilitaire VCM en ligne de commande" dans les liens Référence associée, en fin de rubrique.

## Commande de VCMUtility

```
VCMUtility [<fichier options>] [options]
```

Vous pouvez fournir les options avec le fichier `<fichier options>` spécifié (comme premier paramètre), avec les arguments de la ligne de commande, ou avec les deux. Les arguments de la ligne de commande remplacent toutes les options trouvées dans le `<fichier options>`. Dans le `<fichier options>`, les noms des options commencent à la colonne 1 et excluent le caractère "-"

## Types des commandes de VCMUtility

Cette section contient les types de commandes de `VCMUtility`. Le type de commande par défaut est une nouvelle session VCM.

### Commande entraînant une nouvelle session

Par défaut, chaque exécution de `VCMUtility` commence une nouvelle session VCM, à moins que la commande `Help`, `OPEN`, `Replay`, `Resume`, `Delete` ou `Import` ne soit explicitement émise.

### Commande Help

?

```
Help [<option>]
```

Affiche l'aide de `VCMUtility`. Si vous indiquez une `<option>`, l'aide à cette rubrique est affichée. Par exemple, `Help MMF` affiche de l'aide sur l'option `ManualMergeFiles`.

### Commande Delete

```
Delete <fichier session VCM>
```

Spécifie que la session enregistrée dans le `<fichier session VCM>` spécifié est à supprimer. Tous les fichiers intermédiaires (par exemple, les fichiers des résultats de fusion) et le fichier de session lui-même sont supprimés. Cependant, si la session a été précédemment enregistrée en tant que package de modifications non validées dans la vue cible, l'objet package de modifications n'est pas supprimé.

### Commande Import

```
Import <fichier échange VCM>
```

La commande `Import` est identique à la commande `Resume` sauf que le `<fichier échange VCM>` passé en paramètre doit être un fichier d'échange VCM (`.vcmx`) précédemment créé par une commande `Export`. La session VCM importée reprend là où elle avait été interrompue :

- ◆ La phase de comparaison est exécutée si elle n'a pas déjà été effectuée avec succès.



- ◆ La fusion manuelle est exécutée si `ManualMergeFiles` a été spécifié et s'il existe des conflits de fusion des fichiers.
- ◆ La vue cible "prévisualisation de fusion" est extraite si `CheckoutPreview` a été spécifié et si la validation n'a pas encore été exécutée.
- ◆ Le rapport des différences est généré si `ReportDiffs` a été spécifié et si la validation n'a pas encore été exécutée.
- ◆ La phase de validation est exécutée si `CommitMerge` est `True` et si la validation n'a pas encore été exécutée.
- ◆ Le rapport des mises à jour est généré si `ReportUpdates` a été spécifié et si la validation a été exécutée.

`Export` et `Import` peuvent être utilisés ensemble pour "transporter" une session VCM d'une station de travail à une autre. Par exemple, un utilisateur peut créer une nouvelle session VCM, résoudre tous les conflits et `Exporter` la session. Le fichier d'archive résultant peut ensuite être transféré sur une machine de test où la commande `Import` peut servir, avec l'option `CheckoutPreview` (`CommitMerge` étant défini par **False**), à extraire, à construire et à tester la "prévisualisation" cible de la fusion. Si le test réussit, la machine de test peut ensuite exécuter une commande `Resume` et définir `CommitMerge` par `True`.

**Remarque :** Les sessions ayant été relancées au moyen d'une commande `Resume` ou `Import` ne sont pas automatiquement sauvegardées si elles ne sont pas validées. Si l'option `Save` est spécifiée, la session est enregistrée dans le <fichier session VCM> spécifié. Sinon, c'est le fichier de session VCM spécifié par une commande `Resume` qui est utilisé ; pour une commande `Import`, un nom de fichier de session VCM est généré automatiquement.

### **Commande Open**

`Open <nom Package de modifications>`

Reprend une session VCM précédemment ouverte enregistrée en tant que package de modifications sous le nom donné. Cette option est disponible uniquement sur les serveurs qui prennent en charge les packages de modifications. Le nom spécifié doit être le nom par défaut ou un nom spécifié par l'utilisateur d'un package de modifications enregistré et non validé appartenant au projet (`Project`) et à la vue cible (`TargetView`) spécifiés, qui sont requis. De même, la session ne doit pas être verrouillée par un autre utilisateur, ce qui signifie généralement qu'elle a été déjà ouverte par cet utilisateur.

Pour de plus amples informations, voir les commandes `Name`, `Save`, `Import` et `Resume`.

### **Commande Replay**

`Replay <nom Package de modifications>`

Crée une nouvelle session VCM en "réexécutant" un package de modifications précédemment validé vers une nouvelle vue cible. Cette commande est disponible uniquement quand le serveur prend en charge les packages de modifications. Le package de modifications nommé doit appartenir au projet spécifié par l'option `Project` et à la vue identifiée par l'option `SourceView`. (Comme les packages de modifications validés "appartiennent" à la vue cible qu'ils mettent à jour, la vue cible du package de modifications à réexécuter est toujours la vue source pour la nouvelle session.)

Quand la commande `Replay` est utilisée, la `TargetView` doit être spécifiée, ce qui permet au `MergeType` de la nouvelle session d'être choisi automatiquement en fonction de la relation entre deux vues :

- ◆ Si la vue cible est un enfant de la vue source, une session de refondation est effectuée.
- ◆ Si la vue cible est un parent de la vue source, une session de promotion est effectuée.

- ◆ Sinon, un session de réplication est effectuée.

Alternativement, vous pouvez spécifier le `MergeType` Promotion, auquel cas la vue cible n'est pas nécessaire.

La réexécution d'une session VCM tente d'effectuer les mêmes modifications dans la nouvelle vue cible que celles effectuées dans le package de modifications spécifié. Cela signifie que la portée de la source de la nouvelle session VCM est choisie automatiquement. En conséquence, les options `Include` et `Exclude` ne sont pas autorisées. Dans une session de réexécution, certaines modifications effectuées dans le package de modifications initial peuvent ne pas être possibles dans la nouvelle vue cible (par exemple quand une nouvelle version est déjà présente). Certaines modifications peuvent nécessiter d'être appliquées de façon différente (par exemple, `Move-and-Merge` au lieu de `Merge`), et de nouveaux conflits peuvent apparaître (comme `Merge` au lieu de `Repin`). La session de réexécution ne peut être validée que si aucun conflit non résolu ne se produit.

### **Commande Resume**

`Resume <fichier session VCM>`

Spécifie qu'il faut reprendre la session enregistrée dans le `<fichier session VCM>` et non créer une nouvelle session. Cela sert habituellement à exécuter la phase de validation d'une session précédente pour laquelle seule la phase de comparaison avait été réalisée. Une session ayant déjà été validée peut également être relancée mais uniquement pour générer le rapport des différences. Pour de plus amples informations, voir l'option `Export` et la commande `Import`.

### **Référence associée**

- [Présentation de l'utilitaire VCM en ligne de commande \(VCMUtility\)](#)
- [Options de connexion de VCMUtility](#)
- [Options de session de VCMUtility](#)
- [Options diverses de VCMUtility](#)
- [Exemples VCMUtility](#)
- [Aide-mémoire](#)

## Options de connexion de VCMUtility

Cette section définit les fonctionnalités de `VCMUtility` par rapport aux options de connexion.

### AutoLogon

AL

```
AutoLogon [True] | [False]
```

Si un `<utilisateur>` n'est pas spécifié dans l'option `Server`, `AutoLogon` lance une tentative de connexion avec l'identificateur d'utilisateur et le mot de passe du serveur StarTeam spécifié, tels que stockés dans l'StarTeamUtilitaire Barre d'outils.

### Encryption

Encrypt

En

```
Encryption {NONE | RC4 | RC2_ECB | RC2_CBC | RC2_CFB}
```

Spécifie le niveau de chiffrement de la connexion au serveur. La valeur par défaut est `NONE`. A cause du comportement du SDK, `VCMUtility` rectifiera automatiquement, si nécessaire, le niveau de chiffrement au minimum exigé par le serveur StarTeam.

### PwdFile

PF

```
PwdFile <nom fichier>
```

Spécifie le fichier contenant le mot de passe de connexion. `-PwdFile` remplace le `<mot de passe>` s'il est fourni dans le paramètre `Server`.

### Server

S

```
Server [<utilisateur>[:<mot de passe>]@]<hôte>[:<port>]
```

Spécifie le serveur StarTeam auquel l'utilitaire VCM se connectera.

- ◆ Si `<utilisateur>` et `AutoLogon` ne sont pas spécifiés, l'`<utilisateur>` prend par défaut la valeur "Administrator".
- ◆ Si `<mot de passe>` et `PwdFile` ne sont pas spécifiés, `VCMUtility` demande un mot de passe.
- ◆ Si un `<utilisateur>` ou un `<mot de passe>` contient le caractère ":", ou le caractère "@", ou un espace, il doit être entouré de quotes, simples ou doubles.
- ◆ Si un `<utilisateur>` ou un `<mot de passe>` est entouré de quotes, il peut contenir une quote du même type si elle est précédée d'une barre oblique inversée (\).
- ◆ Si un `<utilisateur>` ou un `<mot de passe>` mis entre guillemets contient un caractère barre oblique inversée, il doit être précédé du caractère d'échappement barre oblique inversée. Par exemple, une double barre oblique inversée dans un élément entre quotes est interprétée comme une barre oblique inversée simple.

- ◆ L'<hôte> du serveur peut être un nom d'hôte ou une adresse IP. L'<hôte> est obligatoire si l'option `Server` est spécifiée.
- ◆ Si l'option `Server` n'est pas spécifiée, l'<hôte> prend par défaut la valeur `localhost`. Si l'option <port> n'est pas spécifiée, elle prend par défaut la valeur 49201.

## UseCA

UCA

```
UseCA {<hôte>:<port> | AutoLocate}
```

Spécifie que les extractions de fichiers doivent tenter d'utiliser un Agent de mise en cache MPX. L'Agent de mise en cache peut être fourni explicitement par le nom ou l'adresse d'un hôte (<hôte> et un numéro de port <port>), ou bien l'Agent de mise en cache le plus proche sur le réseau peut être localisé automatiquement (`AutoLocate`).

## UseServerProfile

USP

```
UseServerProfile [True | False]
```

Si true, spécifie que le nom d'<hôte> spécifié dans l'option `Server` doit être interprété comme un nom de profil de serveur. Les profils de serveur sont stockés dans le fichier `starteam-servers.xml` de l'utilisateur. Un profil de serveur spécifie le nom d'hôte, le numéro de port, le niveau de chiffrement et le paramètre de compression d'un serveur StarTeam. En conséquence, quand `UseServerProfile` est spécifié, l'option `Server` doit être spécifiée, mais ne doit pas contenir de numéro de port, et l'option `Encryption` ne doit pas être spécifiée.

### Référence associée

[Présentation de l'utilitaire VCM en ligne de commande \(VCMUtility\)](#)

[Commandes de VCMUtility](#)

[Options de session de VCMUtility](#)

[Options diverses de VCMUtility](#)

[Exemples VCMUtility](#)

[Aide-mémoire](#)

# Options de session de VCMUtility

Cette section décrit les options de `VCMUtility` qui peuvent être utilisées dans les nouvelles sessions.

Les options de session sont regroupées en deux sections : "Options applicables aux nouvelles sessions" et "Options applicables aux reprises de sessions".

## Options applicables aux nouvelles sessions

Cette section contient les options des sessions de `VCMUtility`.

### ***AutoMergeFiles***

AMF

`AutoMergeFiles [True | False]`

Si `True`, demande une fusion automatique des fichiers qui avaient l'état Fusion dans la phase de comparaison. Lorsqu'une fusion automatique réussit, le fichier résultant devient partie intégrante de la session VCM. Sinon, le fichier résultant est abandonné et les fichiers affectés gardent l'état Fusion non résolue. `AutoMergeFiles` est ignoré pour les sessions de comparaison.

### ***AutoMergeProperties***

AMP

`AutoMergeProperties [True | False]`

Si `True`, demande une fusion automatique des propriétés pour les éléments qui avaient l'état Fusion dans la phase de comparaison. Lorsqu'une fusion automatique de propriétés réussit, l'élément fusionné devient partie intégrante de la session VCM. Sinon, l'élément fusionné est abandonné et les éléments sont marqués comme ayant l'état Fusion de propriétés non résolue. `AutoMergeProperties` est ignoré pour les sessions de comparaison.

### ***BreakLocks***

BL

`BreakLocks [True | False]`

Si `True`, demande une tentative de suspension des verrous posés sur les éléments utilisés dans la phase de comparaison. Suspendre un verrou sur un élément source ou cible n'est obligatoire que si le verrou appartient à un autre utilisateur. La suspension des verrous nécessite une autorisation spéciale et risque donc d'échouer. `BreakLocks` est ignoré pour les sessions de comparaison.

### ***CaseSensitiveFileNames***

CSF

`CaseSensitiveFileNames [True | False]`

Si `True`, les noms de fichiers qui ne diffèrent que par l'utilisation des majuscules/minuscules sont considérés comme différents au moment de l'évaluation de l'option `PreventDuplicateFileNames` et lors de la mise en correspondance des fichiers des vues source et cible.

### ***CheckoutPreview***

Checkout

CP

`CheckoutPreview <fichiers> [<options extraction>]`

Cette option spécifie les fichiers appartenant à la "prévisualisation de fusion" qui doivent être extraits dans l'espace de travail du client. Une "prévisualisation de fusion" est une simulation de la vue cible après sa mise à jour par toutes les modifications issues de la session VCM. `<fichiers>` spécifie les noms et/ou les modèles des noms des fichiers à extraire, à partir des dossiers spécifiés de la vue prévisualisant la fusion. Les `<options extraction>` facultatives contrôlent, par exemple, l'état des fichiers à extraire et l'emplacement où ils seront extraits.

Lorsque `CheckoutPreview` est spécifié, les fichiers sont extraits après la phase de comparaison, après une fusion automatique ou manuelle, mais avant toute validation. L'extraction ne se produit que si la session VCM n'a pas rencontré de conflit de fusion sur le contenu des fichiers. S'il existe des conflits de fusion, une erreur s'affiche et aucune fusion n'est effectuée, indépendamment de l'option `CommitMerge`. S'il n'existe pas de conflit de fusion et si `CommitMerge` vaut `True`, la session VCM est validée après exécution de l'extraction.

Exemple :

```
CheckoutPreview /src/com/acme/*.java +cwf +eol LF +filter CGMIOU +o +ro
+rp C:\BuildDir
```

## **CommitMerge**

Commit

CM

`CommitMerge [True | False]`

Spécifie si les résultats de la session VCM doivent être validés ou non. `False` spécifie que la validation ne sera pas effectuée. Cette option est utilisée pour dédier une session uniquement à la comparaison et à la génération de rapport. `True` spécifie que la validation doit être effectuée s'il n'y a pas de conflit non résolu. `CommitMerge` est ignoré pour les sessions de comparaison.

## **DefaultAction**

DA

`DefaultAction [MergeType <type fusion>] [ItemType <type élément>] <état recherché> <action>`

Spécifie l'`<action>` par défaut pour les éléments comparés qui répondent aux conditions spécifiées dans `<état recherché>`. L'utilitaire VCM se sert d'un tableau de décision établi sur des règles, pour déterminer l'action à exécuter lorsqu'il rencontre des différences entre les éléments de la vue source et de la vue cible. L'option `DefaultAction` permet d'enfreindre les règles par défaut. Cette option peut être spécifiée plusieurs fois pour modifier l'action par défaut de plusieurs différences. Cependant l'ordre dans lequel les actions sont définies est important : si deux actions de remplacement sont applicables à une différence entre éléments trouvée pendant la phase de comparaison, le remplacement spécifié en dernier l'emporte sur le premier.

- ◆ Si `MergeType` est spécifié, `DefaultAction` s'applique uniquement aux sessions VCM du `<type_fusion>` spécifié : Rebase (refondation), Promote (promotion) ou Replicate (réplication).
- ◆ Si `MergeType` n'est pas spécifié, `DefaultAction` s'applique à la session VCM en cours.

Spécifier `DefaultAction` pour un `<type_fusion>` différent de celui de la session en cours permet de spécifier des règles utilisables par différentes sessions VCM dans un même fichier d'options.

Si `ItemType` est spécifié, `DefaultAction` s'applique uniquement aux éléments du `<type_élément>` spécifié : CRs (demandes de modification), Files (fichiers), Folders (dossiers), Requirements (exigences), Tasks (tâches) ou Topics (rubriques). Par défaut, `DefaultAction` s'applique aux éléments de tout type.

L'`<état recherché>` détermine les conditions que doivent respecter les éléments source et/ou cible lors de la comparaison. Un `<état recherché>` consiste en une ou plusieurs définitions de `<condition élément>` source/cible, chacune ayant un `<nom condition>` (par exemple, `source.moved`) et une `<valeur condition>` (`True`, `False` ou `Unspecified`). La `<valeur condition>` est facultative et prend par défaut la valeur `True`. `<état recherché>` représente l'union de toutes les conditions qu'il définit.

L'`<action>` détermine comment gérer les paires d'éléments source/cible dont les différences correspondent à l'`<état recherché>`. L'`<action>` définit seulement l'action à exécuter par défaut sur les éléments respectant les conditions ; la véritable action peut être choisie, après la comparaison, dans le client StarTeam multiplate-forme.

Vous trouverez ci-dessous des exemples de définition de `DefaultAction` :

```
//Lorsqu'un élément source a été déplacé, mais pas l'élément cible,  
//ignorer le déplacement.  
DefaultAction source.moved target.moved false Ignore  
  
//Dans une refondation, si un fichier binaire a été modifié à la fois dans  
//la source et la cible, remplacer la cible par la version source.  
DefaultAction MergeType Rebase  
    items.binaryfile  
    source.modified  
    target.modified  
    Overwrite  
  
//Dans une promotion, si une CR a été déplacée dans les vues source et cible  
//(dans des dossiers différents), déplacer l'élément cible dans le dossier correspondant  
//à l'élément source, mais uniquement si les CR sont dans la même branche.  
DefaultAction MergeType Promote ItemType CR  
    source.moved  
    target.moved  
    items.branched false  
    Move
```

`DefaultAction` est ignoré pour les sessions de comparaison.

## **DefaultComment**

DC

```
DefaultComment <commentaire>
```

Spécifie le commentaire de révision par défaut à utiliser pour les nouvelles révisions d'éléments créées dans la vue cible. `<commentaire>` est une chaîne de texte de forme libre. Dans la valeur du commentaire, chaque occurrence d'une suite d'espaces, y compris les fins de ligne (CR et LF), les blancs et les tabulations, est remplacée par un seul blanc. Par défaut, un commentaire généré automatiquement sert de commentaire de révision par défaut aux nouvelles révisions des éléments. Pour désactiver l'utilisation du commentaire de révision par défaut, spécifiez l'option `DefaultComment` en utilisant une valeur vide.

`DefaultComment` est ignoré pour les sessions de comparaison.

## **Exclude**

Exc

```
Exclude <dossiers>
```

Exclure les dossiers spécifiés de la portée de la source. Seuls les dossiers explicitement spécifiés dans `Include <fichiers>` ou `Include <dossiers>` seront exclus. En conséquence, une option `-Exclude <dossiers>` peut servir à écarter les dossiers sans intérêt de la portée de la source.

Par exemple :

```
//Inclure les CR et les fichiers de tous les dossiers sous /a/b/  
Include /a/b/ +all CRs Files  
  
//Mais exclure les CR du dossier /a/b/c/  
Exclude /a/b/c/ CRs  
  
//Mais si cette CR est dans le dossier /a/b/c/, elle est encore incluse  
Include CR 12345
```

Quel que soit l'ordre des déclarations, les options `Exclude` sont traitées après les options `Include`.

### Sémantique d'Include et d'Exclude

Si aucune option `Include` n'est spécifiée, la portée par défaut de la session VCM est implicitement définie par "tous les fichiers de la vue source". Cela équivaut à spécifier explicitement l'option `include /* +all`. Si au moins une option `Include` est spécifiée, la portée est explicitement limitée aux éléments sélectionnés par les instructions `Include`. Que les portées soient implicites ou explicites, les éléments source sélectionnés sont "taillés" par les éventuelles options `Exclude`.

Toutes les options `Include` et `Exclude` doivent identifier des objets (étiquettes, fichiers, demandes de modification, etc.) de la vue source. Le nom des types sélectionnés peut être au singulier ou au pluriel (`RevLabel`, `CR`, etc.), même si plusieurs valeurs sont fournies.

**Remarque :** Les options `Exclude` sont toujours traitées après les options `Include`, indépendamment de l'ordre des déclarations. De ce fait, `Exclude /src/foo/bar/` suivi de `Include /src/foo/ +all` entraîne l'exclusion du dossier `/src/foo/bar/`.

### Export

`Exp`

`Export <fichier échange VCM>`

L'option `Export` spécifie que toutes les informations sur la session VCM, y compris les fichiers résultant des fusions, sont combinées et enregistrées dans le `<fichier échange VCM>` spécifié. Le nom du fichier d'échange est toujours suffixé de l'extension `.vcmx`. Un fichier d'échange VCM permet de transporter toute la session VCM sur une autre machine ; ensuite, il suffit d'y exécuter une commande `Import` pour reprendre la session. (Pour de plus amples informations, voir la commande `Import`.)

Si le `<fichier échange VCM>` ne contient pas d'informations de chemin, le fichier est enregistré dans le répertoire de base de l'utilisateur (identifié par Java par `user.home`).

**Remarque :** L'option `Export` entraîne toujours la création du fichier d'échange VCM, même lorsque la session elle-même n'est pas enregistrée. Pour de plus amples informations, voir l'option `Save`.

### FixFloatingChildShares

`FFCS`

`FixFloatingChildShares [True | False]`



Spécifie si, dans les opérations de fusion de refondation et de réplcation, chaque élément de la vue cible trouvé comme étant un partagé flottant d'un élément de la vue source doit être "fixé" en étant épinglé. Lorsqu'un élément de la vue cible est un partagé enfant flottant d'un élément source (ce qui implique que l'élément cible n'a pas créé de branche), aucune différence ne sera détectée pendant les sessions VCM entre l'élément source et l'élément cible parce que les modifications apportées à l'élément source "flottent" immédiatement vers l'élément enfant. Le bon usage de VCM suppose de toujours épingler les partagés enfant, afin de contrôler la façon dont les modifications se propagent de la vue source à la vue cible. Cette option permet de "fixer" les éléments enfant flottants trouvés par VCM en les épinglant à la révision de l'élément parent. Spécifier cette option fait baisser les performances en raison du supplément de commandes nécessaires à la vérification de chaque élément cible évalué dans la phase de comparaison.

## **IgnoreMergePoints**

IMP

`IgnoreMergePoints [True] | [False]`

Spécifie si les points de fusion doivent être ignorés pendant la phase de comparaison. Si `True`, les éléments en conflit de fusion utilisent leur point de branchement comme ancêtre commun et non la révision source du dernier point de fusion.

## **Include**

Inc

`Include {<demandes de modification> | <fichiers> | <dossiers> | <éléments de processus> | <exigences> | <étiquettes de révision> | <tâches> | <rubriques> }`

Inclut les éléments spécifiés dans la portée de la source. L'option `Include` peut être fournie plusieurs fois pour permettre l'inclusion de tous les types de sélection. Un seul type de sélection d'éléments (étiquettes de révision, demandes de modification, etc.) doit être spécifié par option `Include`. Le mot-clé désignant le type sélectionné, qui est facultatif pour les fichiers et les dossiers, peut être au singulier ou au pluriel (par exemple, `ProcessItem` ou `ProcessItems`).

Exemples :

```
Include CRs ALL
Include /src/com/*.java +all *.jar +2 *.jspx Buildnumber.h
Include Folders /docs/api/ +all
Include ProcessItem CR 451
Include Reqs 4515 4516
Include RevLabel "Beta Fix 12.413"
Include Topic 14512
Include Task 413
```

## **LockMergeConflicts**

LMC

`LockMergeConflicts {None | Source | Target | Both}`

Spécifie que les éléments dont les conflits ne sont pas résolus doivent être verrouillés de manière exclusive dans la vue Source, dans la vue Target (cible) ou dans Both (les deux) vues. Les verrous sont acquis dans la phase de comparaison. `None` est la valeur par défaut ; elle spécifie de ne pas créer de verrou pour les éléments dont les conflits ne sont pas résolus. Les verrous ne sont appliqués qu'aux éléments source et/ou cible pour lesquels des différences ont été trouvées. Les verrous ne sont pas appliqués aux éléments qui sont comparés et pour lesquels aucune différence n'a été trouvée. De plus, notez que cette option n'est pas affectée par l'option de projet **Demander un verrouillage exclusif lorsque les fichiers sont archivés**, ni par l'option de la station de travail client **Verrouiller**

**les fichiers en mode exclusif lors de l'extraction.** Ces options sont gérées par le moteur VCM lui-même. `LockMergeConflicts` est ignoré pour les sessions de comparaison.

## **ManualMergeFiles**

MMF

`ManualMergeFiles [True | False]`

Si `True`, les outils de fusion des fichiers configurés pour la station de travail sont lancés pour chaque paire de fichiers source/cible trouvée dans l'état Fusion de contenus.

L'option `ManualMergeFiles` peut être utilisée avec `AutoMergeFiles` :

- ◆ Si un conflit de fusion est détecté alors que `AutoMergeFiles` a été demandé, une tentative de fusion automatique est d'abord effectuée.
- ◆ Si le conflit est résolu, le fichier résultant de la fusion est enregistré et la fusion manuelle devient inutile.
- ◆ Si la fusion automatique ne réussit pas ou si `AutoMergeFiles` n'a pas été demandé, et si `ManualMergeFiles` vaut `True`, une fusion manuelle des fichiers est démarrée.

**Remarque :** `ManualMergeFiles` est ignoré (et un avertissement est émis) si la station de travail ne dispose pas d'un outil de fusion manuelle configuré. De même, si l'outil de fusion manuelle ne peut être lancé ou s'il renvoie une condition d'erreur, le fichier concerné reste dans l'état Conflit non résolu. `ManualMergeFiles` est ignoré pour les sessions de comparaison.

## **Match**

`Match [Folder] *{<chemin dossier> to <chemin dossier>}`

Spécifie que pour les opérations de comparaison, le dossier spécifié dans le premier `<chemin dossier>`, qui doit se trouver dans la vue source, doit correspondre au deuxième `<chemin dossier>`, qui doit se trouver dans la vue cible. L'option `Match` est parfois nécessaire pour empêcher des conditions de "correspondance ambiguë", qui peuvent se produire quand une vue non dérivée. Typiquement, l'option `Match` est requise uniquement pour faire correspondre des dossiers racine des vues source et cible. Cependant, d'autres dossiers peuvent être mis en correspondance pour résoudre d'autres conditions de correspondance ambiguë signalées par la phase de comparaison.

Les `<chemin dossier>` source et cible doivent tous les deux commencer et finir par une barre oblique ("/").

Par convention, le dossier racine est représenté par un seul caractère "/". Cela signifie que le nom du dossier racine ne doit pas être fourni dans les chemins de dossiers. Par exemple, si le dossier racine est nommé "StarDraw", le chemin du dossier enfant immédiat "Source Code" est simplement `"/Source Code/`.

Exemples :

```
// Force les dossiers des vues racine source et cible à correspondre.
Match / to /
```

```
// Force le dossier de la vue source "/Source Code" à correspondre au dossier de la vue
cible
//"/Modules/Materials/src".
Match "/Source Code/" to "/Modules/Materials/src/"
```

## **MergeType**

Type

MT

`MergeType {Compare | Rebase | Promote | Replicate}`

Spécifie si on exécute une session de comparaison ou une session de fusion de type refondation, promotion ou réplication. Si seule une vue source ([SourceView](#)) est spécifiée, `MergeType` prend par défaut la valeur Promote. Si seule une vue cible ([TargetView](#)) est spécifiée, `MergeType` prend par défaut la valeur Rebase. Si deux vues source ([SourceView](#)) et cible ([TargetView](#)) sont spécifiées, `MergeType` ne peut être omis. Pour une session de comparaison, les vues source et cible peuvent être les mêmes.

## **Name**

Na

`Name <nom Package de modifications>`

Spécifie le nom du package de modifications associé à la session VCM. Pour les serveurs qui prennent en charge les packages de modifications, un nom est automatiquement choisi au moment où un package de modifications est créé en enregistrant ou en validant la session. Cette option permet d'utiliser un nom spécifique à la place du nom par défaut. Cependant, le nom doit être différent de tous les noms des packages de modifications déjà enregistrés ou validés pour la vue cible, sinon l'enregistrement ou la validation échouera.

Quand l'option `Name` est utilisée en conjonction avec la commande `Open`, le package de modifications ouvert prend le nom de la valeur donnée.

Voir aussi les options `Save` et `CommitMerge`.

## **PostCommitLabel**

PostCL

`PostCommitLabel <étiquette>`

Si la session VCM est validée, l'<étiquette> de *vue* spécifiée est créée dans la vue cible après l'exécution de toutes les mises à jour. L'étiquette reflète les révisions de tous les éléments de la vue cible utilisés dans la phase de comparaison et *modifiés* dans la phase de validation. Cela signifie que l'étiquette contient de nouveaux éléments, de nouvelles révisions d'éléments, des éléments déplacés mais que les éléments supprimés par la validation auront été détachés de l'étiquette. L'étiquette de post-validation est par essence identique à la "prévisualisation de fusion". `PostCommitLabel` est ignoré pour les sessions de comparaison.

Par défaut, une étiquette de vue de post-validation est créée avec un nom par défaut. Pour désactiver l'étiquette de vue de post-validation, spécifiez `PostCommitLabel` avec une valeur vierge (c'est-à-dire " ").

## **PostCommitRevLabel**

PostRL

`PostCommitRevLabel <étiquette>`

Si la session VCM est validée, l'<étiquette> de *révision* spécifiée est créée dans la vue cible et tous les éléments modifiés dans la session VCM, sauf les éléments supprimés, lui sont attachés. En conséquence, l'étiquette contient les éléments ayant été ajoutés, déplacés, réépinglés ou mis à jour, d'une autre manière ou d'une autre, par la session VCM (sauf les suppressions). `PostCommitRevLabel` est ignoré pour les sessions de comparaison.

Par défaut, une étiquette de révision de post-validation n'est pas créée.

## **PreCommitLabel**

PreCL

PreCommitLabel <étiquette>

L'<étiquette> de vue spécifiée est créée dans la vue cible, reflétant la capture utilisée par la phase de comparaison. L'étiquette reflète les révisions de tous les éléments cible utilisés pendant la phase de comparaison. `PreCommitLabel` est ignoré pour les sessions de comparaison.

Par défaut, une étiquette de vue de pré-validation n'est pas créée.

## **PreCommitRevLabel**

PreRL

PreCommitRevLabel <étiquette>

Si la session VCM est validée, l'<étiquette> de révision spécifiée est créée dans la vue cible et tous les éléments non ignorés de la vue cible lui sont attachés, dans leur état "d'avant". C'est-à-dire que les éléments de la vue cible qui doivent être modifiés par la session, sont attachés à l'étiquette de révision avant cette modification. Cela signifie que les éléments à ajouter (par exemple, les éléments partagés) à la vue cible ne **seront pas** attachés, mais les éléments à supprimer le **seront**. `PreCommitRevLabel` est ignoré pour les sessions de comparaison.

Par défaut, une étiquette de révision de pré-validation n'est pas créée.

## **PreventDuplicateFileNames**

PDF

PreventDuplicateFileNames [True | False]

Si `True`, spécifie que le partage d'un nouveau fichier dans la vue cible est interdit s'il crée deux fichiers portant le même nom dans le même dossier.

## **Project**

Pro

Project <projet>

Spécifie le projet à utiliser dans la session VCM. Cette option est obligatoire. Les vues source et cible doivent appartenir au même <projet>. Les noms des projets ne différencient pas les majuscules des minuscules.

## **ReportDiffs**

RD

ReportDiffs [True | False]

Si `True`, entraîne la génération d'un rapport consignait les différences entre éléments trouvées pendant la phase de comparaison. Le rapport des différences est généré dans le répertoire de base de l'utilisateur (identifié par Java par `user.home`) avec le titre suivant :

```
VCMDiffReport-AAAA-MM-JJ_hh-mm-ss.html
```

où `AAAA-MM-JJ` et `hh-mm-ss` sont la date et l'heure en cours dans le fuseau horaire local.

## ReportUpdates

RU

ReportUpdates [True | False]

Si `True`, entraîne la génération d'un rapport consignait toutes les modifications apportées à la vue cible pendant la phase de validation. Le rapport des mises à jour est généré dans le répertoire de base de l'utilisateur (identifié par Java par `user.home`) avec le titre suivant :

```
VCMUpdateReport-AAAA-MM-JJ_hh-mm-ss.html
```

où `AAAA-MM-JJ` et `hh-mm-ss` sont la date et l'heure en cours dans le fuseau horaire local.

`ReportUpdates` est ignoré pour les sessions de comparaison.

## Save

Save [<fichier session VCM>]

Spécifie que la session VCM sera enregistrée. Par défaut, les sessions VCM non validées sont automatiquement enregistrées dans un fichier de session VCM (`.vcms`) avec un nom par défaut dont le format est :

```
<base utilisateur>/VCMSession-AAAA-MM-JJ_hh-mm-ss.vcms
```

où `AAAA-MM-JJ_hh-mm-ss` est la date et l'heure de sauvegarde de la session. Le dossier `<base utilisateur>` est le répertoire de base de l'utilisateur.

Si l'option `Save` est spécifiée avec un nom de `<fichier de session VCM>`, une session non validée sera enregistrée avec ce nom de fichier au lieu du nom par défaut. Si nécessaire, `.vcms` est ajouté au nom. Si le nom de fichier spécifié ne contient pas d'informations de chemin, le fichier de session est enregistré dans le dossier `user.home`.

Le fichier `.vcms` contient les méta-données de la session VCM, mais pas le contenu des fichiers fusionnés. Les contenus des fichiers fusionnés sont stockés dans un dossier temporaire relatif à l'utilisateur et référencés, élément par élément, dans le fichier de session. En conséquence, un fichier `.vcms` ne peut être utilisé que pour reprendre la session VCM sur la même station de travail. (Voir la commande [Resume](#).)

Quand l'option `Save` est spécifiée sans nom de fichier, une tentative d'enregistrement d'une session VCM non validée en tant que package de modifications actif dans la vue cible s'effectue. Le package de modifications est enregistré avec le nom par défaut ou le nom spécifié par l'utilisateur (voir l'option `Name`). Une session VCM enregistrée en tant que package de modifications peut être reprise ultérieurement sur n'importe quelle station de travail à l'aide de l'option `Open`. Cependant, si le serveur ne prend pas en charge les packages de modifications ou si un enregistrement échoue côté serveur, alors la session est enregistrée dans un fichier `.vcms` avec un nom par défaut, comme décrit plus haut.

Quand une validation est effectuée avec succès, l'option `Save` est ignorée. Si le serveur prend en charge les packages de modifications, la session validée crée un package de modifications validé en utilisant le nom par défaut ou le nom spécifié par l'utilisateur (voir l'option `Name`). Si un fichier `.vcms` a été créé précédemment, il est supprimé ainsi que tous les fichiers résultant de la fusion créés par la session VCM.

Voir aussi l'option `Export`.

## SourceLabel

SrcLabel

SL

`SourceLabel <étiquette>`

Demande l'utilisation de la vue source comme une étiquette de vue donnée. Les noms d'étiquettes ne différencient pas les majuscules des minuscules. Une seule des étiquettes `SourceLabel`, `SourceState` et `SourceTime` peut être spécifiée. Si aucune de ces options n'est spécifiée, l'option `SourceTime Now` est utilisée implicitement.

### **SourceState**

`SrcState`

SS

`SourceState <état>`

Demande l'utilisation de la vue source comme un état de promotion de vue donné. Les noms des états de promotion ne différencient pas les minuscules et les majuscules. Une seule des étiquettes `SourceLabel`, `SourceState` et `SourceTime` peut être spécifiée. Si aucune de ces options n'est spécifiée, l'option `SourceTime Now` est utilisée implicitement.

### **SourceTime**

`SrcTime`

ST

`SourceTime {<horodatage> | Now}`

Demande l'utilisation de la vue source comme un horodatage donné. Le mot-clé `Now` provoque l'utilisation d'une capture de l'heure en cours comme horodatage de la configuration. Une seule des étiquettes `SourceLabel`, `SourceState` et `SourceTime` peut être spécifiée. Si aucune de ces options n'est spécifiée, l'option `SourceTime Now` est utilisée implicitement.

### **SourceView**

`Source`

SV

`SourceView <vue>`

Spécifie la vue source à utiliser dans la session VCM. Si plusieurs vues du projet portent le même nom de `<vue>`, un "chemin de vue" découpé par la barre oblique, peut être fourni (par exemple, `VuePrincipale/VueEnfant/VuePetit-enfant`). Si le nom d'une vue contient des barres obliques, il doit être placé entre guillemets.

`SourceView` est facultatif pour les fusions de Refondation ; s'il est spécifié, ce doit être le parent de la vue cible.

**Remarque :** Les noms des vues ne différencient pas les majuscules des minuscules.

### **TargetLabel**

`TgtLabel`

TL

`TargetLabel <étiquette>`

Demande l'utilisation de la vue cible comme une étiquette de vue donnée. `TargetLabel` peut uniquement être utilisé pour les sessions de comparaison. Les noms d'étiquettes ne différencient pas les majuscules des minuscules. Une seule des étiquettes `TargetLabel`, `TargetState` et `TargetTime` peut être spécifiée. Si aucune de ces options n'est spécifiée, l'option `TargetTime Now` est utilisée implicitement.

## TargetState

TgtState

TS

TargetState <état>

Demande l'utilisation de la vue cible comme un état de promotion de vue donné. `TargetState` peut uniquement être utilisé pour les sessions de comparaison. Les noms des états de promotion ne différencient pas les minuscules et les majuscules. Une seule des étiquettes `TargetLabel`, `TargetState` et `TargetTime` peut être spécifiée. Si aucune de ces options n'est spécifiée, l'option `TargetTime Now` est utilisée implicitement

## TargetTime

TgtTime

TT

TargetTime {<horodatage> | Now}

Demande l'utilisation de la vue cible comme un horodatage donné. `TargetTime` peut uniquement être utilisé pour les sessions de comparaison. Le mot-clé `Now` provoque l'utilisation d'une capture de l'heure en cours comme horodatage de la configuration. Une seule des étiquettes `TargetLabel`, `TargetState` et `TargetTime` peut être spécifiée. Si aucune de ces options n'est spécifiée, l'option `TargetTime Now` est utilisée implicitement

## TargetView

Target

TV

TargetView <vue>

Spécifie la vue cible à utiliser dans la session VCM. Si plusieurs vues du projet portent le même nom de <vue>, un "chemin de vue" découpé par la barre oblique, peut être fourni (par exemple, `VuePrincipale/VueEnfant/VuePetit-enfant`). Si le nom d'une vue contient des barres obliques, il doit être placé entre guillemets.

`TargetView` est facultatif pour les fusions de Promotion ; s'il est spécifié, ce doit être le parent de la vue source. Pour les sessions de comparaison, les vues source et cible peuvent être les mêmes.

**Remarque :** Les noms des vues ne différencient pas les majuscules des minuscules.

## Options applicables aux reprises de sessions

Le même fichier d'options pouvant être spécifié avec une commande `Resume`, toutes les options autorisées aux nouvelles sessions le sont aux reprises de sessions. Cependant, la majorité des options, si elles sont re-spécifiées, sont ignorées car elles ne peuvent pas être modifiées une fois que la session a été commencée. Les seules exceptions sont les options citées ci-dessous :

- ◆ Options de connexion : Les informations de connexion (adresse et port du serveur, id utilisateur, mot de passe) ne sont pas stockées dans le fichier de la session VCM, elles doivent être re-spécifiées lors des reprises de sessions. La reprise d'une session échouera si la session n'est pas reconnectée au même serveur StarTeam ou si elle est reprise par un utilisateur différent n'ayant pas les autorisations d'accès aux vues ou aux éléments traités dans la session VCM.
- ◆ `CommitMerge` : Cette option sera habituellement spécifiée avec `True` dans une reprise de session. Cela permet à l'exécution originelle de l'utilitaire VCM de servir uniquement d'exécution de comparaison et à la seconde de servir d'exécution de validation.

- ◆ `ReportDiffs` : Cette option peut être spécifiée dans une reprise de session. Si elle vaut `True`, le rapport des différences est créé avant la phase de validation, si celle-ci a lieu.
- ◆ `ReportUpdates` : Cette option peut être spécifiée dans une reprise de session. Si elle vaut `True` et si la phase de validation réussit, toutes les modifications apportées à la vue cible seront consignées dans un rapport.
- ◆ `CheckoutPreview` : Normalement, si `CheckoutPreview` a été spécifiée dans la session VCM initiale, l'opération d'extraction avec "prévisualisation de fusion" est exécutée dans la reprise de session, avec les mêmes options qu'avant. Cependant, si `CheckoutPreview` est spécifiée dans la reprise de session, elle remplace l'option initiale et provoque l'extraction des fichiers dans la reprise de session selon les nouveaux paramètres.
- ◆ `Description` : Si elle est spécifiée, cette option remplace le texte de description par défaut ou celui fourni précédemment pour le package de modifications. Le nouveau texte de description est utilisé pour la nouvelle révision du package de modifications créée lorsque la session VCM est enregistrée ou validée.
- ◆ `ManualMergeFiles` : Normalement, si `ManualMergeFiles` a été spécifiée dans la session VCM initiale et que la session est enregistrée avec des conflits de fusion de fichiers non résolus, la phase de fusion de fichiers manuelle est effectuée à nouveau lors de la reprise de la session. Cependant, si `ManualMergeFiles` n'a pas été spécifiée dans la session VCM initiale, elle peut être spécifiée comme `True` dans la reprise de session pour appeler la phase de fusion manuelle. Elle peut aussi être spécifiée comme `False` dans une reprise de session pour empêcher la phase de fusion manuelle.
- ◆ `PostCommitLabel`, `PostCommitRevLabel`, `PreCommitLabel` et `PreCommitRevLabel` : Si l'une quelconque de ces options d'étiquette est spécifiée dans une option de reprise, elle remplace la valeur précédente relative à l'étiquette correspondante. Quand une option d'étiquette est définie par un caractère blanc (" "), l'option d'étiquette en question est désactivée et ne sera pas créée au cours de phase de validation.

#### Référence associée

[Présentation de l'utilitaire VCM en ligne de commande \(VCMUtility\)](#)  
[Commandes de VCMUtility](#)  
[Options de connexion de VCMUtility](#)  
[Options diverses de VCMUtility](#)  
[Exemples VCMUtility](#)  
[Aide-mémoire](#)



## Options diverses de VCMUtility

Cette section définit les options diverses de `VCMUtility` non sauvegardées avec les sessions de comparaison/fusion des vues.

### NetMon

NM

`NetMon [True | False]`

Active la fonctionnalité de surveillance du réseau du SDK. Chaque commande émise par `VCMUtility` au serveur StarTeam est enregistrée dans la fenêtre console (mais pas dans le fichier journal de `VCMUtility`).

### Time

T

`Time [True | False]`

Entraîne l'affichage d'informations temporelles pour chaque phase de la session VCM exécutée. Les informations temporelles sont écrites à la fois dans la fenêtre console et dans le fichier journal de `VCMUtility`.

### Verbose

vb

V

`Verbose [True | False]`

Entraîne l'affichage d'informations supplémentaires de diagnostic et de progression sur la console (sortie standard) et dans le fichier journal de `VCMUtility` pendant l'exécution.

### Référence associée

- [Présentation de l'utilitaire VCM en ligne de commande \(VCMUtility\)](#)
- [Commandes de VCMUtility](#)
- [Options de connexion de VCMUtility](#)
- [Options de session de VCMUtility](#)
- [Exemples VCMUtility](#)
- [Aide-mémoire](#)

## Exemples VCMUtility

Cette rubrique présente des exemples de l'utilisation de [VCMUtility](#) pour divers types de fusions.

### Refondation de Hello World

Vous trouverez ci-dessous les options de l'équivalent "Hello World" d'une exécution de [VCMUtility](#) pour une refondation :

```
Type      Refondation
Project   Hello
Target    World
```

### Refondation automatique

Le fichier d'options ci-dessous exécute la même refondation que l'exemple précédent, mais il effectue, si possible, la validation et fournit un rapport détaillé sur les résultats :

```
Type              Rebase
Project           Hello
Target            World
CommitMerge       True
LockMergeConflicts Both

// Toutes ces options sont définies par True :
AutoMergeFiles
BreakLocks
ReportDiffs
ReportUpdates
```

Tous les fichiers sont fusionnés automatiquement, en termes de contenu comme de propriétés. Les fichiers dont le conflit est impossible à résoudre sont verrouillés à la fois dans la vue source et dans la vue cible. Les conflits de verrouillage existants sont démêlés autant que possible. Si aucun conflit non résolu n'est rencontré, la session est validée. Les détails à la fois de la phase de comparaison (différences) et de la phase de validation (mises à jour) sont consignés dans un rapport. Si la validation est réussie, tous les fichiers temporaires de la session VCM sont supprimés.

### Promotion par étiquette de vue : Comparaison uniquement

Les options ci-dessous effectuent, en comparaison uniquement, une promotion des fichiers et des demandes de modification (CR) à partir d'une étiquette de vue, en enregistrant la session sous un nom de fichier de session spécifique :

```
// Paramètres de connexion
Serveur           MyUserId@ProdServer:4000
PwdFile           MyPassword.txt

// Type de la fusion et configuration de la vue
Type              Promotion
Project           StarDraw
Source            "Beta Release"
SrcLabel          Build-4.0_142
```

```
// Sélectionner tous les fichiers et demandes de modification comme éléments source
include          /* +all
include          / +all CRs

// Comparer uniquement, créer un rapport et enregistrer sous un nom de fichier de session
spécifique
CommitMerge     False
save            Build-4.0_142-Promote
ReportDiffs

// Options diverses
AutoMergeFiles  True
AutoMergeProperties  False // garder en tant que conflits et fusionner manuellement
LockMergeConflicts  Target
```

## Promotion par étiquette de vue : Fusion

La ligne de commande de l'utilitaire VCM ci-dessous reprend la session enregistrée dans l'exemple précédent et la valide, en supposant qu'aucun nouveau conflit ne soit survenu.

```
VCMUtility -resume Build-4.0_142-Promote -CommitMerge -ReportUpdates
```

### Référence associée

[Présentation de l'utilitaire VCM en ligne de commande \(VCMUtility\)](#)

[Commandes de VCMUtility](#)

[Options de connexion de VCMUtility](#)

[Options de session de VCMUtility](#)

[Options diverses de VCMUtility](#)

[Aide-mémoire](#)

# Aide-mémoire

Syntaxe de `VCMUtility` en ligne de commande : `VCMUtility [<fichier options>] [*<option>]`

Dans le `<fichier options>`, chaque `<option>` doit commencer à la colonne 1 et peut se poursuivre sur les lignes d'après si ces lignes commencent par un espace ou un caractère de tabulation. Lorsque vous saisissez des options sur la ligne de commande, chaque `<option>` doit être précédée d'un "-".

## Options

Le tableau ci-dessous liste les options de `VCMUtility` en ligne de commande accompagnées de leur syntaxe.

---

### Commande/Option

---

```
<option>
  <commande> | <option connexion> | <option session> | <option diverse>
```

---

```
<commande>
  {{Help | H | ?} [<rubrique aide>]} |
  {Delete <fichier session VCM>} |
  {Import <fichier archive VCM> |
  {Open <nom Package de modifications>} |
  {Replay <nom Package de modifications>} |
  {Resume <fichier session VCM>}
```

---

```
<option connexion>
  {{AutoLogon | AL} [True | False]} |
  {{Encryption | Encrypt | En} {None | RC4 | RC2_ECB | RC2_CBC | RC2_CFB}} |
  {{PwdFile | PF} <nom fichier>} |
  {{Server | S} [<utilisateur>[:<mot de passe>]@<hôte>[:<port>]} |
  {{UseCA | UCA} {<hôte>:<port> | AutoLocate}} |
  {{UseServerProfile | USP} [True | False]}
```

---

```
<option session>
  {{AutoMergeFiles | AMF} [True | False]} |
  {{AutoMergeProperties | AMP} [True | False]} |
  {{BreakLocks | BL} [True | False]} |
  {{CaseSensitiveFileNames | CSF} [True | False]} |
  {{CheckoutPreview | check-out | CP} <fichiers> [<options extraction>]} |
  {{CommitMerge | Commit | CM} [True | False]} |
  {{DefaultAction | DA} [MergeType <type fusion>] [ItemType <type élément>] <état
recherché> <action>} |
  {{DefaultComment | DC} <commentaire>} |
  {{Description |} <description>} |
  {{Exclude | Exc} <dossiers>} |
  {{Export | Exp}<fichier archive VCM>} |
  {{FixFloatingChildShares | True | False}} |
  {{IgnoreMergePoints | IMP} [True | False]} |
  {{Include | Inc} {<demandes de modification> | <fichiers> | <dossiers> | <éléments
de processus> |
  <exigences> | <étiquettes de révision> | <tâches> | <rubriques>}} |
  {{LockMergeConflicts | LMC} {None | Source | Target | Both}} |
  {{ManualMergeFiles | MMF} [True | False]} |
  {{Match [Folder] *{<chemin dossier> to <chemin dossier>}} |
  {{MergeType | Type | MT} {Compare | Rebase | Promote | Replicate}} |
  {{Name | Na} <nom Package de modifications>} |
```

```
{{PostCommitLabel | PostCL} <étiquette>} |
{{PostCommitRevLabel | PostRL} <étiquette>} |
{{PreCommitLabel | PreCL} <étiquette>} |
{{PreCommitRevLabel | PreRL} <étiquette>} |
{{PreventDuplicateFileNames | PDF} [True | False]} |
{{Project | Pro} <projet>} |
{{ReportDiffs | RD} [True | False]} |
{{ReportUpdates | RU} [True | False]} |
{Save [<fichier session VCM>]} |
{{SourceLabel | SrcLabel | SL} <étiquette>} |
{{SourceState | SrcState | SS} <état>} |
{{SourceTime | SrcTime | ST} {<horodatage> | Now}} |
{{SourceView | Source | SV} <vue>} |
{{TargetLabel | TgtLabel | TL} <étiquette>} |
{{TargetState | TgtState | TS} <état>} |
{{TargetView | Target | TV} <vue> |
```

```
<option diverse>
  {{NetMon | NM} [True | False]} |
  {{Time | T} [True | False]} |
  {{Verbose | Vb | V} [True | False]} |
```

## Autres éléments syntaxiques

Le tableau ci-dessous liste les autres éléments syntaxiques par ordre alphabétique :

### Autres éléments syntaxiques

```
<action>
  Delete | DeleteAndReverseShare | Fail | Ignore | Merge | Move | MoveAndMerge |
  MoveAndRepin | NeedsReview | Overwrite | Repin | RepinAndMove |
  ReverseShare | Share
```

```
<demandes de modification>
  {CR | CRs | ChangeRequests} {ALL | *<N° CR>}
```

```
<nom Package de modifications>
  {Un nom constitué d'un ou de plusieurs caractères}
```

```
<options extraction>
  [+cwf] [+eol {on | off | cr | lf}] [+filter {CGIMOU}] [+o] [+ro]
  [+rp <chemin dossier travail>]
```

```
<nom condition>
  items.binaryfile | items.branched | items.samecontent | source.childshare |
  source.deleted | source.floating | source.modified | source.moved |
  source.present | source.rootbranch | target.childshare | target.deleted |
  target.floating | target.modified | target.moved | target.present |
  target.parentdeleted | target.rootbranch
```

---

```
<valeur condition>
  True | False | Unspecified
```

---

```
<fichiers>
  [File | Files] {ALL | *{<modèle nom fichier>} [+<profondeur>]}
```

---

```
<chemin du dossier>
{Une barre oblique suivie d'une série facultative de noms de dossiers, chacun se terminant par une barre oblique}
```

---

```
<dossiers>
  [Folder | Folders] {ALL | *{<chemin dossier>} [+<profondeur>] *{<type élément>}}
```

---

```
<condition élément>
  <nom condition> [<valeur condition>]
```

---

```
<type élément>
  {ChangeRequest | CR | ChangeRequests | CRs} |
  {File | Files} |
  {Folder | Folders} |
  {Requirement | Req | Requirements | Reqs}
  {Task | Tasks}
  {Topic | Topics}
```

---

```
<état recherché>
  *{<condition élément>}
```

---

```
<élément processus>
  ProcessItems *{[View <vue>] CR <N° CR> |
                 [View <vue>] Req <N° Exig> |
                 [View <vue>] Task <N° Tâche>}
```

---

```
<exigences>
  {Requirement | Req | Requirements | Reqs} {ALL | *{<N° Exig>}}
```

---

```
<étiquettes de révision>
  RevLabels *{<étiquette>}
```

---

```
<tâches>
  {Task | Tasks} {ALL | *{<N° Tâche>}}
```

---

```
<horodatage>
  Exemples de formats :
  "11/03/06 1:32 PM"
  "11 mars 2006 1:32:38 PM"
  "11.03.06 1:32:38 PM PST"
  "samedi 11 mars 2006 1:32:38 PM PST"
```

---

<rubriques>  
{Topic | Topics} {ALL | \*<N° Rub>}

---

<fichier échange VCM>  
Un nom de fichier .vcms}

---

<fichier session VCM>  
Un nom de fichier .vcms}

---

## Référence associée

[Présentation de l'utilitaire VCM en ligne de commande \(VCMUtility\)](#)

[Commandes de VCMUtility](#)

[Options de connexion de VCMUtility](#)

[Options de session de VCMUtility](#)

[Options diverses de VCMUtility](#)

[Exemples VCMUtility](#)

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<demandes de modification>](#)

[<options extraction>](#)

[<fichiers>](#)

[<dossiers>](#)

[<élément de processus>](#)

[<exigences>](#)

[<étiquettes de révision>](#)

[<tâches>](#)

[<horodatage>](#)

[<rubriques>](#)

# Syntaxe des options composées de VCMUtility

Les éléments syntaxiques composés suivants sont utilisés dans les options de VCMUtility.

## Dans cette section

### [<action>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <action>.

### [<options extraction>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <options extraction>.

### [<demandes de modification>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <demandes de modifications>.

### [<fichiers>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <fichiers>.

### [<dossiers>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <dossiers>.

### [<type élément>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <état recherché>.

### [<état recherché>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <état recherché>.

### [<élément de processus>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <élément de processus>.

### [<exigences>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <exigences>.

### [<étiquettes de révision>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <étiquettes de révision>.

### [<tâches>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <tâches>.

### [<horodatage>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <horodatage>.

### [<rubriques>](#)

Décrit la syntaxe de l'option composée de VCMUtility, <rubriques>.



## <action>

Spécifie l'action à exécuter pour une certaine différence entre éléments source/cible. Une <action> est un des mnémoniques suivants :

Mnémonique	Description
Delete	Supprime l'élément cible.
DeleteAndReverseShare	Equivalent à un Delete (supprimer) suivi d'un ReverseShare (intervertir le partage).
Fail	Synonyme de NeedsReview (voir plus loin).
Ignore	Ne rien faire.
MarkResolved	Crée un point de fusion qui ne fait que marquer les éléments source et cible comme étant résolus.
Merge	Fusionne les éléments source et cible.
Move	Déplace l'élément cible dans le dossier équivalent à l'élément source.
MoveAndMerge	Equivalent à un Move (déplacer) suivi d'un Merge (fusionner).
MoveAndOverwrite	Equivalent à un Move (déplacer) suivi d'un Overwrite (écraser).
MoveAndRepin	Equivalent à un Move (déplacer) suivi d'un Repin (réépingler).
NeedsReview	Force une révision avant la validation. C'est-à-dire que la validation n'est pas autorisée lorsque cette action est sélectionnée. Les différences entre les éléments marqués par cette action sont impossibles à concilier, et donc l'action doit être changée en quelque chose d'autre.
Overwrite	Ecrase la cible par le contenu de la source.
Repin	Change la révision à laquelle est épinglée la cible pour correspondre à l'élément source (réépingler).
ReverseShare	Déplace l'élément source dans la vue cible et le re-partage avec la vue source (intervertir le partage).
Share	Partage l'élément source dans la vue cible.

**Remarque :** Toutes les <action>s ne sont pas valides pour toutes les différences qui existent entre les éléments. Par exemple, Delete n'est pas recevable lorsque l'élément cible a déjà été supprimé.

### Référence associée

[Syntaxe des options composées de VCMUtility](#)  
[<demandes de modification>](#)  
[<options extraction>](#)  
[<fichiers>](#)  
[<dossiers>](#)  
[<état recherché>](#)  
[<élément de processus>](#)  
[<exigences>](#)  
[<étiquettes de révision>](#)  
[<tâches>](#)  
[<horodatage>](#)  
[<rubriques>](#)  
[Aide-mémoire](#)

## <options extraction>

La section suivante décrit la syntaxe de l'option composée <options extraction> de VCMUtility.

```
[+cwf] [+eol {on | off | cr | lf | crlf}] [+filter {CGIMOU}] [+o] [+ro] [+rp <chemin dossier travail>]
```

Spécifie les options d'extraction différentes des options par défaut. Les options d'extraction disponibles sont celles fournies aux utilitaires en ligne de commande StarTeam (stcmd) et BCO (Bulk Check-out), sauf que le nom de ces options doit être préfixé par un signe '+'. Les options disponibles sont détaillées ci-après.

### +cwf

Demande la création des dossiers de travail pour tous les dossiers spécifiés, même s'ils ne contiennent aucun des fichiers à extraire lors de l'exécution. Seuls les dossiers visibles sont créés.

### +eol <option eol>

Demande la conversion au format spécifié de tous les délimiteurs de fin de ligne trouvés dans les fichiers texte. La valeur `on` de <option eol> utilise le format EOL configuré sur le client. `off` interdit toute conversion EOL. `cr`, `lf` et `crlf` entraînent la conversion de chaque EOL en retour-chariot, saut de ligne ou paire retour-chariot/saut de ligne, respectivement. Notez que les fichiers texte ayant un format EOL "fixe" sont toujours convertis au format spécifié.

### +filter

```
+filter {CGIMOU}
```

Spécifie l'état des fichiers à extraire : **C**urrent (à jour), **merGe** (fusion), **m**issing (manquant), **M**odified (modifié), **O**ut-of-date (obsolète) ou **U**nknown (inconnu). Plusieurs indicateurs d'états peuvent se combiner. Si `+filter` n'est pas spécifié, le filtre par défaut est **IO** (**M**issing (manquant) et **O**ut-of-date (obsolète)). Si les fichiers **Merge**, **Merge** ou **Unknown** sont inclus sans l'option `+o`, un avertissement est émis pour chaque fichier correspondant et celui-ci n'est pas extrait.

### +o

Spécifie d'inclure, en plus des fichiers **Missing** et **Out-of-date**, les fichiers dont l'état est **Modified**, **Merge** ou **Unknown**. De plus, tous les fichiers sont écrasés sans avertissement. Si `+filter` est spécifié également, seuls les fichiers spécifiés sont extraits.

### +ro

Définit chaque fichier comme étant en lecture seule après l'extraction. Par défaut, les fichiers extraits sont accessibles en lecture et en écriture.

### +rp

Spécifie le dossier de travail racine de la "prévisualisation de fusion". Les fichiers sont extraits dans des dossiers de travail enfant relatifs au <chemin dossier travail>

## Référence associée

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<demandes de modification>](#)

[<fichiers>](#)

[<dossiers>](#)

[<état recherché>](#)

[<élément de processus>](#)

[<exigences>](#)

[<étiquettes de révision>](#)

[<tâches>](#)

[<horodatage>](#)

[<rubriques>](#)

[Aide-mémoire](#)

## <demandes de modification>

<demandes de modifications> {CR | CRs | ChangeRequests} {ALL | \*<CR #>}

Spécifie toutes les demandes de modifications de la vue ou certaines d'entre elles indiquées par le numéro de demande de modifications. `CRs` et `ChangeRequests` sont synonymes ; les formes au singulier sont également acceptées.

### Référence associée

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<options extraction>](#)

[<fichiers>](#)

[<dossiers>](#)

[<état recherché>](#)

[<élément de processus>](#)

[<exigences>](#)

[<étiquettes de révision>](#)

[<tâches>](#)

[<horodatage>](#)

[<rubriques>](#)

[Aide-mémoire](#)

## <fichiers>

```
[File | Files] {ALL | *{<modèle nom fichier>} [+<profondeur>]}
```

Spécifie tous les fichiers de la vue ou un ensemble de fichiers spécifiques donné sous la forme d'une liste de noms et/ou de modèles de noms de fichiers, chacun étant associé à une <profondeur> de dossiers optionnelle. Le mot-clé `File` (ou `Files`) est facultatif sauf si le mot-clé `All` est utilisé. Un <modèle nom fichier> peut être un nom de fichier particulier (par exemple, `foo.java`), un modèle de noms de fichiers (par exemple, `*.java`), ou un nom de fichier ou un modèle de noms de fichiers contenant le chemin du dossier (par exemple, `(/src/com/acme/foo.java)` ou `/src/com/acme/*.java`).

### Utilisation

Les chemins des dossiers doivent utiliser les barres obliques ; une seule barre oblique (/) est synonyme du dossier racine. (Par conformité avec les autres utilitaires de StarTeam, le nom du dossier racine, qui correspond habituellement au nom de la vue, ne doit **pas** être fourni dans les noms des chemins.)

- ◆ Si un nom de fichier ou un modèle est fourni sans le chemin du dossier, le dossier impliqué est celui du précédent paramètre <modèle nom fichier>.
- ◆ Si le premier paramètre <modèle nom fichier> ne contient pas le chemin du dossier, le dossier impliqué est le dossier racine.
- ◆ Si elle est fournie, la <profondeur> de dossiers spécifie le nombre de niveaux de dossiers enfant, sous le dossier spécifié, à inclure ; cela peut être un nombre ou le mot-clé **All**.
- ◆ Si un nom de fichier ou un modèle de noms de fichiers contient des espaces, il doit être placé entre guillemets.

### Exemples

Voici des exemples de l'utilisation de <fichiers> :

```
// tous les fichiers de la vue
include Files ALL

//foo.java et bar.java du dossier /src/com/acme
include /src/com/acme/foo.java bar.java

//tous les fichiers java du dossier /src/com/acme et au dessous
include /src/com/acme/*.java +all

// tous les fichiers .txt du dossier racine, tous les fichiers .zip du premier niveau
// des dossiers enfant, et un fichier readme.txt particulier
include *.txt *.zip +1 /docs/acme/readme.txt
```

## Référence associée

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<demandes de modification>](#)

[<options extraction>](#)

[<dossiers>](#)

[<état recherché>](#)

[<élément de processus>](#)

[<exigences>](#)

[<étiquettes de révision>](#)

[<tâches>](#)

[<horodatage>](#)

[<rubriques>](#)

[Aide-mémoire](#)

## <dossiers>

```
[Folder | Folders] {ALL | *{<chemin dossier>} [+<profondeur>] * [<type élément>]}
```

Spécifie tous les chemins des dossiers de la vue ou des chemins de dossiers spécifiques, en indiquant optionnellement une profondeur de dossiers et des types d'éléments spécifiques. Le mot-clé `Folder` (ou `Folders`) est facultatif sauf si le mot-clé `All` est utilisé.

### Utilisation

Un `<chemin dossier>` doit commencer et finir par une barre oblique (`/src/com/`). Si elle est fournie, la `<profondeur>` spécifie le nombre de niveaux de dossiers enfant, sous le dossier spécifié, à inclure ; cela peut être un nombre ou le mot-clé `All`.

- ◆ Si un chemin de dossier contient des espaces, il doit être placé entre guillemets.
- ◆ Si aucun paramètre `<type élément>` n'est fourni, seuls les fichiers du ou des dossiers spécifiés sont inclus. Sinon, tous les éléments des types spécifiés sont inclus.

Les types d'éléments reconnus sont `CRs`, `Files`, `Folders`, `Tasks`, `Topics` et `Requirements` (au singulier ou au pluriel).

### Exemples

Voici des exemples de l'utilisation de `<dossier>` :

```
// tous les dossiers de la vue
include folders ALL

//tous les fichiers du seul dossier /src/com/acme/
include /src/com/acme/

//tous les fichiers et toutes les tâches du dossier /src/ et au dessous
include /src/ +all files tasks

// toutes les CR du dossier "/triage/" et tous les fichiers des dossiers
// enfant situés à deux niveaux en dessous de "/PR docs/"
include /triage/ CRs "/PR docs/" +2
```

Par convention, le dossier racine est représenté par un seul caractère `"/`. Cela signifie que le nom du dossier racine ne doit pas être fourni dans les chemins de dossiers. Par exemple, si le dossier racine est nommé "StarDraw", le chemin du dossier enfant immédiat "Source Code" est simplement `/Source Code/`.

## Référence associée

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<demandes de modification>](#)

[<options extraction>](#)

[<fichiers>](#)

[<état recherché>](#)

[<élément de processus>](#)

[<exigences>](#)

[<étiquettes de révision>](#)

[<tâches>](#)

[<horodatage>](#)

[<rubriques>](#)

[Aide-mémoire](#)



## <type élément>

<type élément>

Spécifie un type d'élément. Les valeurs autorisées sont [ChangeRequest](#) (ou [CR](#)), [File](#), [Folder](#), [Requirement](#) (ou [Req](#)), [Task](#) et [Topic](#). Les noms des types d'élément ne distinguent pas les majuscules des minuscules et peuvent être au pluriel.

### Référence associée

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<demandes de modification>](#)

[<options extraction>](#)

[<fichiers>](#)

[<dossiers>](#)

[<élément de processus>](#)

[<exigences>](#)

[<étiquettes de révision>](#)

[<tâches>](#)

[<horodatage>](#)

[<rubriques>](#)

[Aide-mémoire](#)

## <état recherché>

\*<condition élément>

Définit un ensemble de conditions qui s'appliquent aux différences entre éléments source et éléments cible. <état recherché> représente l'union de toutes les conditions <condition élément> qu'il définit. Chaque <condition élément> prend la forme :

<nom condition> [<valeur condition>]

## <nom condition>

Les paramètres <nom condition> valides et leur signification sont les suivants :

<nom condition>	Signification
items.binaryfile	Indique si l'un des éléments concernés est un fichier binaire.
items.branched	Indique si les éléments source et cible sont dans des branches différentes de l'arborescence des versions des objets.
items.samecontent	Indique si les éléments source et cible ont les mêmes propriétés modifiables par l'utilisateur et, pour les fichiers, le même contenu de données.
source.childshare	Indique si l'élément source est un partagé enfant de l'élément cible.
source.deleted	Indique si l'élément concerné a été supprimé de la vue source.
source.floating	Indique si l'élément source a une configuration flottante.
source.modified	Indique si l'élément concerné a été modifié dans la vue source.
source.moved	Indique si l'élément concerné a été déplacé dans la vue source.
source.present	Indique si l'élément concerné est présent dans la vue source.
source.rootbranch	Indique si l'élément source est la branche racine de son arborescence de partage.
target.childshare	Indique si l'élément cible est un partagé enfant de l'élément source.
target.deleted	Indique si l'élément concerné a été supprimé de la vue cible.
target.floating	Indique si l'élément cible a une configuration flottante.
target.modified	Indique si l'élément concerné a été modifié dans la vue cible.
target.moved	Indique si l'élément concerné a été déplacé dans la vue cible.
target.parentdeleted	Indique si le dossier de l'élément cible a été supprimé.
target.present	Indique si l'élément concerné est présent dans la vue cible.
target.rootbranch	Indique si l'élément cible est la branche racine de son arborescence de partage.

## <valeur condition>

Les paramètres <valeur condition> valides et leur signification sont les suivants :

<valeur condition>	Signification
True	La condition est vraie pour le ou les éléments auxquels elle est applicable.
False	La condition est fausse pour le ou les éléments auxquels elle est applicable.

La <valeur condition> est facultative et prend par défaut la valeur `True`. Pour tout <état recherché>, toutes les conditions non spécifiées possèdent l'état `Unspecified`.

Une <condition élément> doit être définie par `True` ou par `False` pour entraîner la recherche de la condition correspondante dans les différences réelles entre éléments.

Une condition peut être définie par `Unspecified`, par exemple, afin de supprimer la condition des critères de recherche, pour un essai, sans supprimer la condition dans le fichier d'options.

**Remarque :** Certaines conditions s'excluent mutuellement : si elles sont définies ensemble, elles ne correspondront jamais à aucune différence réelle entre éléments. Par exemple, un élément source ne peut pas être, à la fois, présent (`source.present=true`) et supprimé (`source.deleted=true`).

### Référence associée

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<demandes de modification>](#)

[<options extraction>](#)

[<fichiers>](#)

[<dossiers>](#)

[<élément de processus>](#)

[<exigences>](#)

[<étiquettes de révision>](#)

[<tâches>](#)

[<horodatage>](#)

[<rubriques>](#)

[Aide-mémoire](#)

## <élément de processus>

```
ProcessItems *{[View <vue>] CR <N° CR> | [View <vue>] Req <N° Exig> | [View <vue>] Task <N° Tâche>}
```

Spécifie le jeu d'éléments de processus (demandes de modification, tâches et/ou exigences) à inclure. Spécifier un élément de processus conduit à inclure également les éléments qui lui sont liés dans la vue source. Le mot-clé `ProcessItems` peut être au singulier. Les noms complets `ChangeRequest` et `Requirement` peuvent être utilisés à la place de `CR` et `Req`, respectivement.

Par défaut, tout élément de processus spécifié doit appartenir à la vue source. Cependant, le préfixe optionnel `View <vue>` peut servir à sélectionner un élément de processus dans une autre vue que la vue source. Lorsqu'un élément de processus appartenant à une autre vue que la vue source est inclus, l'élément de processus n'est **pas** inclus dans la portée de la source, mais les éléments qui lui sont liés dans la vue source le sont. La révision spécifique de chaque élément de la vue source lié à l'élément de processus est incluse.

### Exemples :

```
// Inclure la CR 451 de la vue source et ses éléments liés
include ProcessItem CR 451

//Inclure les éléments de la vue source liés à la tâche 909
//Inclure l'exigence 518, les deux appartenant à la vue "Triage"
//include ProcessItem Vue Triage Tâche 909
                        View Triage Requirement 518
```

**Remarque :** Si le nom de la vue contient des espaces, il doit être placé entre guillemets ("`Release 4.3`"). Si plusieurs vues du projet portent le même nom de vue, celui-ci peut être fourni comme un chemin découpé par la barre oblique ("`Apps/Releases/Release 4.3`").

### Référence associée

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<demandes de modification>](#)

[<options extraction>](#)

[<fichiers>](#)

[<dossiers>](#)

[<état recherché>](#)

[<exigences>](#)

[<étiquettes de révision>](#)

[<tâches>](#)

[<horodatage>](#)

[<rubriques>](#)

[Aide-mémoire](#)

## <exigences>

{Reqs | Requirements} {ALL | \*{<N° Exig>}}

Spécifie des exigences individuelles par leur numéro. *Reqs* et *Requirements* sont synonymes ; les formes au singulier sont également acceptées.

### Référence associée

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<demandes de modification>](#)

[<options extraction>](#)

[<fichiers>](#)

[<dossiers>](#)

[<état recherché>](#)

[<élément de processus>](#)

[<étiquettes de révision>](#)

[<tâches>](#)

[<horodatage>](#)

[<rubriques>](#)

[Aide-mémoire](#)

## <étiquettes de révision>

RevLabels \*<étiquette>

Spécifie tous les éléments attachés à chacune des étiquettes de révision spécifiées (<étiquette>). Le mot clé `RevLabels` peut être au singulier. Les noms des étiquettes de révision ne différencient pas les majuscules des minuscules.

### Référence associée

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<demandes de modification>](#)

[<options extraction>](#)

[<fichiers>](#)

[<dossiers>](#)

[<état recherché>](#)

[<élément de processus>](#)

[<exigences>](#)

[<tâches>](#)

[<horodatage>](#)

[<rubriques>](#)

[Aide-mémoire](#)

## <tâches>

Tasks {ALL | \*{<N° Tâche>}}

Inclut les tâches individuelles spécifiées par leur numéro. Le mot clé `Tasks` peut être au singulier.

### Référence associée

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<demandes de modification>](#)

[<options extraction>](#)

[<fichiers>](#)

[<dossiers>](#)

[<état recherché>](#)

[<élément de processus>](#)

[<exigences>](#)

[<étiquettes de révision>](#)

[<horodatage>](#)

[<rubriques>](#)

[Aide-mémoire](#)

## <horodatage>

Un <horodatage> doit utiliser un des formats reconnus par Java pour les chaînes de date et heure.

- ◆ Les formats de dates suivent les conventions locales (par exemple, 3/11/06 sera interprété March 11, 2006 aux Etats-Unis.)
- ◆ Les seconds sont facultatives (par exemple, 1:32 et 1:32:00 sont identiques).
- ◆ L'indicateur AM/PM est obligatoire.
- ◆ L'indicateur de fuseau horaire est facultatif ; s'il est omis, le fuseau horaire local est utilisé.
- ◆ S'il est fourni, le jour de la semaine est ignoré.

### Exemples :

```
"11/03/06 1:32 PM"  
"11 mars 2006 1:32:38 PM"  
"11.03.06 1:32:38 PM PST"  
"samedi 11 mars 2006 1:32:38 PM PST"
```

### Référence associée

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<demandes de modification>](#)

[<options extraction>](#)

[<fichiers>](#)

[<dossiers>](#)

[<état recherché>](#)

[<élément de processus>](#)

[<exigences>](#)

[<étiquettes de révision>](#)

[<tâches>](#)

[<rubriques>](#)

[Aide-mémoire](#)



## <rubriques>

Topics {ALL | \*{<N° Rub>}}

Inclut les rubriques individuelles spécifiées par leur numéro. Le mot clé `Topics` peut être au singulier.

### Référence associée

[Syntaxe des options composées de VCMUtility](#)

[<action>](#)

[<demandes de modification>](#)

[<options extraction>](#)

[<fichiers>](#)

[<dossiers>](#)

[<état recherché>](#)

[<élément de processus>](#)

[<exigences>](#)

[<étiquettes de révision>](#)

[<tâches>](#)

[<horodatage>](#)

[Aide-mémoire](#)

## Index

- All, 109
- Bulk Check-out, utilitaire
  - options de la ligne de commande, 9
- Checkout Trace, utilitaire
  - opérations en ligne de commande, 25
- False, 80
- Fichiers journal de VCM
  - VCMUtility, 79
- Packages de modifications
  - VCMUtility, 79
- Vault Verify
  - ligne de commande, 26
- VCMUtility
  - Présentation, 76
  - <action>, 105
  - <demandes de modification>, 108
  - <dossiers>, 111
  - <élément de processus>, 116
  - <état recherché>, 114
  - <étiquettes de révision>, 118
  - <exigences>, 117
  - <fichiers>, 109
  - <horodatage>, 120
  - <options extraction>, 106
  - <rubriques>, 121
  - <tâches>, 119
  - <type élément>, 113
  - abréviations, 78
  - Aide-mémoire, 100
  - AutoLogon, 83
  - AutoMergeFiles, 85
  - AutoMergeProperties, 85
  - BreakLocks, 85
  - CaseSensitiveFileNames, 85
  - CheckoutPreview, 85
  - codes de sortie, 78
  - commande, 76 80
  - commandes, 80
  - CommitMerge, 86
  - Conventions syntaxiques, 76
  - DefaultAction, 86
  - DefaultComment, 87
  - Encryption, 83
  - Exclude, 87
  - exemples, 98
  - Export, 88
  - fichier d'options, 77
  - FixFloatingChildShares, 88
  - IgnoreMergePoints, 89
  - Include, 89
  - LockMergeConflicts, 89
  - ManualMergeFiles, 90
  - Match, 90
  - MergeType, 91
  - NetMon, 97
  - nom du package de modifications, 91
  - Options applicables aux reprises de sessions, 95
  - Options booléennes, 78
  - options de connexion, 83
  - options de session, 85
  - options diverses, 97
  - paramètres de la ligne de commande, 77
  - PostCommitLabel, 91
  - PostCommitRevLabel, 91
  - PreCommitLabel, 92
  - PreCommitRevLabel, 92
  - PreventDuplicateFileNames, 92
  - Project, 92
  - promotion par étiquette de vue, 98 99
  - PWDFFile, 83
  - refondation automatique, 98
  - ReportDiffs, 92
  - ReportUpdates, 93
  - Save, 93
  - Server, 83
  - sources en entrée, 77
  - SourceTime, 94
  - SourceView, 94
  - SourceState, 94
  - SrcLabel, 93
  - TargetLabel, 94
  - TargetState, 95
  - TargetTime, 95
  - TargetView, 95
  - Time, 97
  - Type de commandes Delete, 80
  - Type de commandes entraînant une nouvelle session, 80
  - Type de commandes Help, 80
  - Type de commandes Import, 80
  - Type de commandes Open, 81
  - Type de commandes Replay, 81
  - Type de commandes Resume, 82
  - types de commandes, 80
  - UseCA, 84 84
  - valeurs des options et Unicode, 77
  - Verbose, 97