Borland® StarTeam® 2009

StarTeam コマンドライン ツール ヘルプ

Borland®

Borland Software Corporation
8310 N Capital of Texas Hwy, Bldg 2, Ste 100
Austin, Texas 78731 USA
www.borland.com/jp

2009 年 6 月
PDF

コマンド ライン ツール

コマンド ライン ツール

このセクションでは、さまざまなコマンド ライン オペレーション ツールについて説明します。

このセクションの内容

参照

ここでは、参照情報を提供します。

このセクションの内容

[コマンドライン操作]
コマンドライン操作に関する参照情報を提供します。

[クライアントのコマンドライン操作]
このセクションでは、クライアントのコマンドライン操作に関する参照情報を提供します。

[VCM コマンドライン ユーティリティ]
ビュー比較/マージのコマンドライン ユーティリティ(VCMUtility)について説明します。

コマンドライン操作

コマンドライン操作に関する参照情報を提供します。

このセクションの内容

一括チェックアウト ユーティリティのコマンドライン オプション

ここでは、BCO について説明します。

- 構文
- コマンドライン オプション
- BCO の使用例

構文

BCO では、以下の構文を使用します。

```
-p "projectSpecifier" [-pwdfile "filePath"] [-autoLogon] [-cwf] [-is] [-rp "folderPath" | -fp "folderPath" ] [-cmp] [-dryrun] [-vb] [-useCA] [-encrypt encryptionType] [-cfgl "labelName"| -cfgp "stateName" | -cfgd "asOfDate"] [-filter "fileStatus"] [-o] [-ro] [-ts] [-fs] [-eol on | off | cr| lf| crlf ] [-exclude <pattern> | @<pattern file>] [-netmon] [-t] [-h | -help] [files...]
```

オプション

以下の表は、BCO で使用できるオプションについて説明しています。

オプション	説明
-autoLogon	-p オプションでユーザー名が指定されていない場合、Toolbar Utility で保存されているユーザー ID とパスワードを使って、指定されたサーバーへのログインが試みられます。これが可能なのは、Windows オペレーティング システムでだけです。
-cfgd	指定した日付/時刻のものとしてビューを構成します。日時指定の例を以下に示します。 ■ "12/29/01 10:52 AM" ■ "December 29, 2001 10:52:00 AM PST" ■ "Monday, December 29, 2001 10:52:00 AM PST"
-cfgl	指定されたラベルを使用してビューを構成します。-cfgl、-cfgp、または -cfgd がない場合、BCO は現在の構成をビューに使用します。
-cfgp	指定されたプロモーション状態を使用してビューを構成します。
-cmp	ワークステーションとサーバー間で送られるすべてのデータを圧縮し、到着時に解凍します。このオプションを省略すると圧縮は行われません。 圧縮は、クライアントとサーバーが低速の接続を介して通信する場合に便利です。圧縮を使用するかどうかを決定するために、小さいテストケースで試してみることをお勧めします。データの圧縮と解凍に費やされる時間が発生しても、低速接続を介して長時間かけて非圧縮データを送るより良いかどうかという観点で検討しなければなりません。
-cwf	ベース フォルダに対する作業フォルダを作成します。-is オプションを指定している場合は、ベース フォルダのすべてのサブフォルダに対する作業フォルダも作成されます。実行中にチェックアウトするファイルを持たないフォルダに対しても、作業フォルダが作成されます。表示されているフォルダだけが作成されます。
-csf	コマンドの実行によって、-p オプションで指定したフォルダを基本となる StarTeam のフォルダにマップする場合に、-csf を指定すると、StarTeam のフォルダの名前の大文字小文字が区別されるようになります(このオプションは、フォルダ内のファイル名の大文字小文字の扱いには適用されません)。たとえば、-csf を指定すると、doc という名前の StarTeam フォルダと Doc という名前の StarTeam フォルダ

	は別フォルダと見なされます。このオプションを指定しないと、どちらのフォルダも同じ「doc」フォルダと見なされます。 デフォルトでは、StarTeam フォルダの名前は大文字小文字が区別されません。 -csf を使用してもしなくてもファイル名があいまいな場合はエラーとなります。たとえば、-csf を使用する際、Doc と doc フォルダの両方がある場合 2 つのフォルダの名前はあいまいです。-csf を使用しない場合、スペルが同じフォルダ名はあいまいです。
-dryrun	ファイルをチェックアウトせずに、-dryrun を指定しなければチェックアウトされることになるファイルの一覧を表示します。このパスは、ファイルのコピー先となる作業フォルダのパスです。-vb を一緒に使用すると、チェックアウトが実行された場合の詳細な情報を得ることができます。
-encrypt RC4 | RC2_ECB | RC2_CBC | RC2_CFB	ワークステーションとサーバー間で送られるすべてのデータを暗号化し、到着時に復号します。このオプションを省略すると暗号化は行われません。暗号化すれば、安全が確保されていないネットワーク回線上で、ファイルなどのプロジェクト情報が第三者に不正に読み取られないようにすることができます。 完全な構文は以下のとおりです。-encrypt encryptionType BCO は、以下の種類の暗号化をサポートしています。 *RC4*: RSA RC4 ストリーム暗号(高速) *RC2_ECB*: RSA RC2 ブロック暗号(Electronic Codebook) *RC2_CBC*: RSA RC2 ブロック暗号(Cipher Block Chaining) *RC2_CFB*: (Windows プラットフォームのみ)RSA RC2 ブロック暗号(Cipher Feedback) これらの暗号化レベルは高速なものから順番に並べられています。低速な暗号化タイプほど、安全性は高くなります。
-eol { on | off | cr | lf | crlf }	改行コードを自動的に変換します。 このオプションを -on にすると、テキスト ファイルがサーバーのリポジトリからワークステーションの作業フォルダへと転送されるときに、コマンドを実行しているプラットフォームに合わせて Java VM で指定されたとおりに改行コードが変換されます。 このオプションを指定しないと、デフォルトでは、BCO は改行コードの変換を実行しません。-eol を全く使用しないと、デフォルトと同じ結果になります。 改行文字(cr、lf、または crlf)を指定すると、テキスト ファイルがサーバーのリポジトリからワークステーションの作業フォルダに転送されるときに、指定した改行変換が行われます。 改行コードは、Windows プラットフォームでは復帰/改行(crlf)、UNIX プラットフォームでは改行(lf)です。 たとえば、リポジトリから取得したファイルを UNIX システム上の作業ファイルと比較する場合(リポジトリにはテキスト ファイルが crlf ファイルとして保存されている場合)、このオプションを設定します。
-exclude <pattern> | @<pattern file>	名前または親フォルダが指定されたパターンに一致するファイルを除外します。パターンには正確なファイル名またはフォルダ名を指定できます。また、ワイルドカード文字('*.class' など)を使用することもできます。フォルダ名を指定するには、パターン名の直前にスラッシュ('/bin' など)を入れます。各 -exclude で単一のパターンを指定することができ、それを繰り返すことができます。あるいは、与えられた <pattern file>(先頭に @ が付く)の個々の行にパターンを 1 つ以上指定することもできます。
files...	コマンドで使用するファイルを名前またはファイル名パターン("*.c"など)で指定します。UNIX のセマンティック規約の方が Windows よりも明確なので、すべてのオプションは UNIX の規約を使用して解釈されます。このため、"すべてのファイル" を指定するには、"*.*" ではなく "*" を使います。パターン "*.*" は、"名前と拡張子を持つすべてのファイル" を意味します。たとえば、"star*.*" は

starteam.doc と starteam.cpp を見つけますが、starteam は見つけません。これらすべてを検索するには、"star*" を使用します。

このオプションを省略するとデフォルトは "*" です。

すべてのファイルを指すために、"*" ではなく * を使用すると、UNIX シェルはそれを一連のアイテムに展開し、オプション グループとして bco コマンドに渡してしまいます。たとえば、作業ファイルなしのファイルをチェックアウトしている場合、これにより問題が生じる可能性があります。混乱を避けるために、"*" を使用することをお勧めします。

複数のファイル パターンを使用する場合、各パターンをそれぞれ引用符で囲む必要があります。たとえば、"*.bat" "*.c" は使用できますが "*.bat *.c" は使用できません。

注: このオプションは、プラットフォームに関係なく必ず引用符で囲むことをお勧めします。ただし、その理由はプラットフォームごとに異なります。Windows プラットフォームでは、空白を含むファイル名とフォルダ名は正しく解釈されません。UNIX プラットフォームでは、シェルがオプションを展開し、展開して作られたアイテムのリストがクライアントに渡されます。しばしば、これによって予期しない結果が生じます。このような結果を避けるために、このオプションは常に引用符で囲むことをお勧めします。UNIX シェルによるオプションの展開が必須の場合にのみ、引用符を省略してください。いずれのプラットフォームでも、バッチ スクリプトが原因不明で失敗する場合、引用符の省略が原因となっている可能性があります。

ファイルの指定では、以下の特殊文字を使用できます。

* は、任意の文字列(空白の文字列を含む)と一致します。たとえば、"*" は、拡張子の有無に関係なくあらゆるファイル名と一致します。"xyz*" は、"xyz"、"xyz.cpp"、"xyzutyfj" のいずれとも一致します。

? は、任意の 1 文字と一致します。たとえば、"a?c" は、"abc" とは一致しますが、"ac" とは一致しません。

[...] は、左右のかぎ括弧に囲まれた文字のいずれか 1 つと一致します。

2 つの文字の間にハイフン(-)を入れると、その範囲内のいずれかの文字と一致します。

左かぎ括弧([)の直後に感嘆符(!) またはキャラット(^)を置くと、括弧内の文字とは一致しません。カギ括弧で囲まれていない任意の文字と一致します。たとえば、"x[a-d]y" は、"xby" とは一致しますが、"xey" とは一致しません。"x[!a-d]y" は、"xey" とは一致しますが、"xby" とは一致しません。

ハイフン(-)または右カギ括弧( ])そのものと一致させるには、カギ括弧内の最初または最後の文字として含めます。

パターン内でアスタリスク(*)、疑問符(?)、または左カギ括弧([)そのものと一致させるには、その直前にエスケープ文字(¥)を入れる必要があります。

-filter	1 つまたは複数の文字から成る文字列を指定します。各文字が、ファイル ステータスを表します。この文字列には、スペースやその他のホワイト スペースは入れないでください。現在、指定されたステータスを持っているファイルだけがチェックアウトされます。*ビュー外*のファイルをチェックアウトすることはできません。 ステータスを表すために使用できる文字は、次のとおりです。C(最新)、M(変更済み)、G(マージ)、O(古いリビジョン)、I(作業ファイルなし)、U(不明)。 C、M、G、O、U のいずれかをしてする場合は、-o も指定して、チェックアウト処理を強制しなければなりません。-o オプションを指定しない場合、BCO は、C、M、G、O、または、U で表されたステータスを持つファイルをスキップしますが、ログ ファイルには警告が出力されます。
-fp "folderPath"	指定されたフォルダの作業フォルダ(作業ディレクトリ)をオーバライドします。これは、フォルダの代替作業パスを設定するのと同じ結果となります。 これにより、このフォルダで指定されたものとは別の作業フォルダを使用することができますが、もっとも重要なのは、プラットフォーム間の互換性を提供するという用途です。たとえば、UNIX と Windows システムでは、ドライブとディレクトリ パス

	名の指定方法に互換性がありません。"D:¥MYPRODUCT¥DEVELOPMENT¥SOURCE" というパスは、Windows プラットフォームでは解釈されますが、UNIX プラットフォームでは解釈されません。 自分のプラットフォームがプロジェクトで指定されたパスを解釈できない場合、このオプションを使用して作業パスを定義します。 引用符の前に置かれた円記号(¥)は、エスケープ文字として解釈されます。したがって、次の例はエラーとなります。 bco -p "xxx" -fp "C:¥" "*" これは次のように解釈されます。 bco -p "xxx" -fp "C:" *" これを避けるため、"C:¥" の最後の文字を次のようにエスケープします。 bco -p "xxx" -fp "C:¥¥" "*" または、-fp パスが "C:¥orion¥" のようにルート フォルダで終わらない場合は、次のようにしてこの問題を回避します。 bco -p "xxx" -fp "C:¥orion" "*" また、UNC ネットワーク パスの場合、さらに追加でエスケープ文字が必要な点に注意してください。例: bco -p "xxx" -fp "¥¥¥server¥path" 完全な構文は以下のとおりです: -rp "folderPath"。 フォルダは Windows 用語で、これがユーザー インターフェイスに表示されます。UNIX プラットフォームでは、ディレクトリが正しい用語です。
-fs	チェックアウト後にファイル ステータスが記憶されないようにします。これらのファイルの以降のステータス値は、正しくなく不定になります。ファイルのステータスが重要ではない場合に、このオプションを使用します。たとえば、いつも作業フォルダを削除してからビルドのためにファイルをチェックアウトしている場合、ファイルが存在しないので、そのステータスは問題になりません。 このファイル ステータスは、たとえ後から update-status コマンドを使ったとしても既知にはなりません。-fs オプションを指定せずにチェックアウトを強制すれば、正しいステータスを持つ最新のファイルが得られます。
-h -help	コマンドライン オプションについての情報を表示します。
-is	コマンドをすべての子フォルダに再帰的に適用します。このオプションを省略すると、コマンドは指定したフォルダにのみ適用されます。
-netmon	SDK NetMonitor の情報をコンソール ウィンドウに出力します。NetMonitor はサーバー コマンドの統計を表示します。この章の次のセクションの BCO の使用例を参照してください。
-o	通常であればチェックアウトされないステータスを持つファイルを強制的にチェックアウトします。該当するステータスは、*更新済み*、マージ、*不明*です。
-p	使用するビューやフォルダ、およびサーバーにアクセスするために必要なユーザー名とパスワードを指定します。 完全な構文は以下のとおりです。 -p "userName:password@hostName:portNumber/ projectName/[viewName/][folderHierarchy/]" たとえば、以下のように指定します: -p "JMarsh:password@orion:49201/ StarDraw/ StarDraw/SourceCode/"。 ユーザー名を省略すると、現在のユーザー名が使われます。この例では、ユーザー名は JMarsh です。

	パスワード を省略すると、パスワードの入力を要求されます。ユーザーがパスワードを入力するとき、文字は画面上に表示されません。この例では、パスワードは password です。 ホスト名を省略した場合のデフォルトは localhost です。この例では、ホスト名は orion です。 ポート番号は必須です。この例では、デフォルトのポート番号 49201 を使用しています。 プロジェクト名は必須です。この例では、プロジェクト名は StarDraw です。 ビューの指定には、ビュー階層を使用します。ビュー名の区切り文字にはコロン(:)を使用します。ビュー階層には、必ずルート ビューを含めます。たとえば、"StarDraw:Release 4:Service Packs" は、*StarDraw* ルート ビューの孫で *Release 4* ビューの子の *Service Packs* ビューを使用することを意味します。ビュー名を省略すると、ルート ビュー(デフォルト)が使われます。プロジェクト内にその名前を持つビューが 1 つしかない場合は、ビュー名だけを指定することもできます。(ただし、後から同じ名前のビューが作られる可能性もあるので、このような指定方法は推奨できません。) この例では、ビュー名は *StarDraw* です。これは *StarDraw* プロジェクトのルート ビューなので、省略することも可能です。 フォルダの指定には、フォルダ階層を使用します。フォルダ名の区切り文字にはスラッシュ(/)を使用します。フォルダ階層には、ルート フォルダを含めてはなりません。ファイルがビューのルート フォルダにある場合はフォルダ階層を省略します。たとえば、ビューのルート フォルダが *StarDraw* でファイルへの階層が "StarDraw/SourceCode/Client" ならば、単に "SourceCode/Client" を使用します。 このオプションで使用するパラメータ(ユーザー名、ビュー名、パスワード、プロジェクト名、フォルダ名など)に、区切り文字として使用される文字が含まれている場合、その文字にはパーセント記号(%)で始まる 16 進数コードを使用します。たとえば、パスワードの中に「@」が表れる場合、「%40」で置き換える必要があります。 ":" には、"%3a" を使用。 "/" には、"%2f" を使用。 "@" には、"%40" を使用。 "%" には、"%25" を使用。 UNIX および他のオペレーティング システムでは、一部の特殊文字の先頭に円記号 "¥" または別のエスケープ文字を指定する必要があります。-p オプションでは、そのような文字を 16 進数コードで置き換えることができます。たとえば、UNIX で "¥<" とする代わりに "%3c" を使用することができます。 空白には "%20" を使用。 "<" には、"%3c" を使用。 ">" には、"%3e" を使用。
-pwdfile	ユーザーのパスワードが格納されているファイルのパスを指定します。このオプションは -p オプションで使われるパスワードよりも優先されます。これによって、コマンドライン上で他人にパスワードを見られるのを防止できます。このファイルは、UTF-8 形式で保存されていなければなりません。 完全な構文は以下のとおりです。 -pwdfile "filePath"
-ro	この操作の後、作業ファイルを読み取り専用にします。このオプションを指定しないと、ファイルは操作の前と同じ状態のままとなります。通常は、 -ro を使用して、自分がロックしていないファイルを自分自身が編集することを防ぎます。-ro を指定しないと、ファイルは読み取り/書き込み可能になります。
-rp	ビューのルート フォルダ用の作業フォルダまたは作業ディレクトリを指定または上書きします。これは、ビューの代替作業パスを設定するのと同じ結果となります。

	補足情報については、この表の中の -fp の説明を参照してください。 完全な構文は以下のとおりです。 -rp "folderPath"
-t	チェックアウトのボリュームとタイミングに関する統計情報を表示します。
-ts	各作業ファイルのタイム スタンプをチェックアウトの時刻に設定します。このオプションを指定しないと、ファイルには、そのファイルのチェックイン リビジョンと同じタイム スタンプが与えられます。
-useCA	StarTeamMPX Cache Agent を使用して、ファイルのチェックアウトを試みます。 完全な構文は以下のとおりです。 -useCA host:port | autolocate 構文 host:port では、使用する Cache Agent のホスト名(または IP アドレス)とポート番号を指定します。 代わりに、autolocate を指定して、ネットワーク上最も近い Cache Agent を自動的に探すよう指定することもできます。autolocate を使用するには、StarTeam Server で MPX が有効になっていることが必要です。
-vb	出力を詳細にします。各ファイルがチェックアウトされるときに、その情報が表示されるようになります。このフォルダのパスは、作業フォルダのパスではなく、フォルダのパスです。

BCO の使用例

次の例は、BCO を使って、ルート フォルダ *StarDraw*(*StarDraw* プロジェクトの *StarDraw* ビュー内)の子である *Source Code* フォルダにあるすべてのファイルを強制的にチェックアウトします。

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

次に示すのは、-netmon オプションを使用した BCO コマンドと、Net Monitor によって表示された出力です。

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

出力サンプル:

```
StarTeam BulkCheckOut Utility version 9.0.xxx
Copyright (c) 2006 Borland Software Corporation. All rights reserved.
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
```

starteamserver コマンドのパラメータ

ここでは、starteamserver コマンドのオプションについて、それぞれの使用例とともにアルファベット順に説明します。

-access Key

組み合わせるオプション: -serial。

参照: -serial、-license、-eval。

ライセンスされたバージョンとしてサーバーを登録します。このオプションは -serial オプションと組み合わせて使用します。サーバーを最初に起動するとき、ライセンスされたバージョンまたは評価版コピーのどちらかとしてアプリケーションを登録する必要があります。シリアル番号とアクセス キーの組み合わせ、または評価期間を延長するための評価キーが必要な場合は、 http://www.borland.com/us/company/how-to-buy.html までお問い合わせください。

例:

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

-all

組み合わせるオプション: -start、-stop、-restart。

-start(または-restart)または -stop オプションと組み合わせて使用します。-start -all オプションは、starteam-server-configs.xml ファイルでステータスが「準備完了」になっているすべてのサーバー構成を起動します。-stop -all オプションは、ステータスが「実行中」であるすべてのサーバー構成を停止します。

例:

```
starteamserver -stop -all
```

-attach "AttachmentsPath"

組み合わせるオプション: -start、-restart。

サーバー構成の添付ファイルのパスを指定します。

サーバー構成を最初に起動するとき、システムは RepositoryPath で指定されたパスの下に Attachments サブフォルダを作成し、サーバー構成が使用するデータベースにこのフォルダへのパスを格納します。Attachments フォルダの場所を変更する場合は、コマンドラインから -start(または -restart)および -attach オプションを使用すれば、データベース内の添付ファイルのパスを変更できます。

添付ファイルのパスの変更は、[StarTeam Server の構成] ダイアログの [全般] タブで行うこともできます。添付ファイルの新しいパスは、次回にサーバー構成を起動したときに有効になります。

例:

```
starteamserver -start MyServer -attach "c:¥My Server¥Attachments"
```

-autorecover

組み合わせるオプション: -start。

参照: -stoponerrors。

-autorecover オプションは、検証処理の間に修復が必要になった場合に最低限の修復を試みるようサーバーに指示します。

例:

```
starteamserver -start MyServer -autorecover
```

-dsn "DataSourceName"

組み合わせるオプション: -new、-edit、-start、-restart。

参照: -t、-p、-u。

データベース接続情報を指定します。既存の ODBC データ ソース名(DSN)を指定します。

リリース 5.1 および 5.2 では、$ORACLE_HOME/network/admin/tnsnames.ora に保存されている Oracle ネット サービス名を使って Oracle データベースにアクセスしていました。現在はこうではありません。

DBServerName に指定する値は starteam-serverconfigs.xml ファイルに格納されます。データベース接続情報は、以下の方法で確認または変更できます。

- コマンドラインから -view および -edit オプションを使用します。
- [StarTeam 管理] ダイアログから **StarTeam Server の構成** ダイアログ ボックスを表示して [データベース] タブを選択する。
- [サーバー管理] ダイアログから [サーバーの構成 | プロパティ] ダイアログ ボックスを表示して [データベース] タブを選択する。

変更内容は、次回にサーバー構成を起動したときに有効になります。

例:

```
starteamserver -edit MyServer -dsn MyServerDSN
```

-edit "ConfigurationName"

組み合わせるオプション: -name、-dsn、-u、-p。

指定されたサーバー構成のセッション オプションを編集します。編集できるオプションは -name, -dsn, -u, -p です。サーバー構成が実行中の場合、編集を行う前にサーバー構成を停止する必要があります。

例:

```
starteamserver -edit MyServer -name Portable -dsn RemoteServer
-u StarTeamAdmin
```

-eval Number

参照: -serial、-access、-license。

サーバーの評価版コピーの評価期間を延長します。サーバーを最初に起動するとき、ライセンスされたバージョンまたは評価版コピーのどちらかとしてアプリケーションを登録する必要があります。シリアル番号とアクセス キーの組み合わせ、

または評価期間を延長するための評価キーが必要な場合は、 http://www.borland.com/us/company/how-to-buy.html までお問い合わせください。

例:

```
starteamserver -eval 01234567890
```

-help

すべてのコマンド オプションを説明するメッセージを表示します。

例:

```
starteamserver -help
```

-licenses

参照: -serial、-access、-eval。

ライセンスおよび登録情報を表示します。アプリケーションの評価版コピーを実行している場合、そのことを知らせるメッセージが表示されます。そうでない場合は、シリアル番号が表示されます。

例:

```
starteamserver -licenses
```

-list

starteam-server-configs.xml ファイルに定義されているサーバー構成と各構成のステータスを一覧表示します。特定の時点におけるサーバー構成のステータスは、「準備完了」、「起動中」、「実行中」、「停止中」のいずれかです。

例:

```
starteamserver -list
```

サーバーは次のようなメッセージを表示します。

```
Configuration Status MyServer Ready StarDrawRepository Running
Portable Ready
```

-name "ConfigurationName"

組み合わせるオプション: -edit、-start、-restart。

サーバー構成の名前を変更します。このオプションは -edit オプションと組み合わせて使用します。サーバー構成の新しい名前は、次回にサーバー構成を起動したときに有効になります。

例:

```
starteamserver -edit MyServer -name NewTeamServer
```

-new "ConfigurationName"

指定した名前と設定を持つ新しいサーバー構成に、DefaultHive という名前のハイブを作成します。このサーバー構成は Native-II データ保管庫を使用します。このオプションは、サーバー管理 ツールのメニューで **新規作成...** を選択し、ウィザードを使って新規サーバー構成を作成した場合と同じ結果になります。

多くのオプションが、-new との組み合わせでのみ指定することができます。次のものがあります: -c、-r、-t。

例:

```
starteamserver -new NewServer1 -r "c:¥new server¥" -t 1 -dsn NewServerDSN
-u Admin -p password
```

-p "DBUserPassword"

組み合わせるオプション: -new、-edit、-start、-restart。

参照: -dsn、-p、-t、-u。

データベースにアクセスするために使われるパスワードを指定します。DBUserPassword に指定した値は starteam-server-configs.xml ファイルに格納されます。指定するパスワードが、データベースのユーザー名に対して正しいものであることを確認してください。パスワードとユーザー名は、コマンドラインから -view および -edit オプションを使用して確認または変更できます。変更内容は、次回にサーバー構成を起動したときに有効になります。

例:

```
starteamserver -edit MyServer -u JodyK -p password
```

-q

コマンドを無出力モード(システムは画面出力を表示しない)で実行します。

例:

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

-r "RepositoryPath"

組み合わせるオプション: -new

新しいサーバー構成のリポジトリ パスを指定します。指定したリポジトリ パスが存在しない場合、このサーバー構成を最初に起動するときにシステムが適切なフォルダを作成します。

RepositoryPath に指定する値は starteam-serverconfigs.xml ファイルに格納されます。リポジトリ パスの確認は、コマンドラインから -view オプションを使用するか、またはサーバー管理ツール内の StarTeam Server の構成ツールの **全般** タブで行うことができます。

警告: サーバー構成のリポジトリ パスとしてサーバーのホーム フォルダ(ディレクトリ)を使用しないでください。サーバー構成が起動しなくなります。

例:starteamserver -new NewServer1 -r "c:¥new server¥" -t 1 -dsn NewServerDSN -u Admin -p password

-remove "ConfigurationName"

指定されたサーバー構成を starteam-server-configs.xml ファイルから削除します。

例:

```
starteamserver -remove MyServer
```

-restart "ConfigurationName"

指定されたサーバー構成を停止して再起動します。このオプションは、サーバー構成に対して変更を行った後で、その変更を有効にしたい場合に使用します。サーバー構成の再起動が失敗した場合は、サーバー ログ ファイルで詳細を確認してください。

サーバー構成を再起動して、その構成の複数のオプションを同時に変更できます。-restart オプションと組み合わせて使用できるオプションには、次のものがあります: -all、-attach、-dsn、-name、-p、-tcpip、-u。-all オプションと特定の構成名を同時に指定することはできません。

例:

```
starteamserver -restart MyServer -tcpip StarTeamTCPIP -u SuperUser
-p SuperUserPassword
```

-serial Number

参照: -access、-license、-eval。

ライセンスされたバージョンとしてサーバーを登録します。このオプションは -access オプションと組み合わせて使用します。サーバーを最初に起動するとき、ライセンスされたバージョンまたは評価版コピーのどちらかとしてアプリケーションを登録する必要があります。シリアル番号とアクセス キーの組み合わせ、または評価期間を延長するための評価キーが必要な場合は、 http://www.borland.com/us/company/how-to-buy.html までお問い合わせください。次の例のシリアル番号とアクセス番号は、実際のシリアル番号とアクセス番号に置き換えます。

例:

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

-start "ConfigurationName"

参照: -all、-stop。

指定されたサーバー構成を起動します。構成が起動すると、starteam-server-configs.xml ファイル内のこのサーバー構成のエントリで、Status=Running および PID=nnn に更新されます。ここでの nnn は、実際の PID 番号に置き換わります。

サーバー構成を起動して、その構成の複数のオプションを同時に変更できます。-start オプションと組み合わせて使用できるオプションには、次のものがあります: -attach、-dsn、-name、-p、-tcpip、-u。

例:

```
starteamserver -start MyServer -tcpip StarTeamTCPIP -u SuperUser
-p SuperUserPassword
```

-stop "ConfigurationName"

参照: -all、-start。

指定されたサーバー構成を停止します。サーバー構成が停止すると、starteam-server-configs.xml ファイル内のこのサーバー構成のエントリで **Status=Ready** および PID=0 に更新されます。

例:

```
starteamserver -stop MyServer
```

注: Enterprise Advantage ユーザーの場合: サーバーをサービスとして実行し、依存するサービスとして Notification Agent を実行している場合、まず Notification Agent サービスを停止しないとチームのサーバーを停止できません。

-t DBType

組み合わせるオプション: -new

参照: -dsn、-p、-u。

データベースのタイプを指定します。このオプションは新しいサーバー構成の作成時にのみ使用できます。データベースの種類を指定する値として、以下のいずれかを使用します。

2 = Microsoft SQL Server または SSE

3 = Oracle

DBType に指定した値は starteam-server-configs.xml ファイルに格納されます。データベースの種類は以下の方法で確認できます:

- コマンドラインから -view オプション。
- サーバー管理ツールの **StarTeam Server の構成** タブの **データベース** タブ。
- サーバー管理ツールの <server configuration=""> **プロパティ** ダイアログ ボックスの **データベース** タブ。

例:

```
starteamserver -new NewServer1 -r "c:¥new server¥" -t 1 -dsn NewServerDSN
-u Admin -p password
```

-tcpip Endpoint | up[:Endpoint] | down[:Endpoint]

組み合わせるオプション: -start、 -restart

TCP/IP(ソケット)プロトコル用のエンドポイントを指定します。また、プロトコルを有効または無効にします。有効にするには up を、無効にするには down を使用します。up または down の後にコロンとエンドポイントを続けて指定すると、プロトコルを有効または無効にすると同時にエンドポイントを設定できます。

エンドポイントとして指定した値は、このサーバー構成が使用するデータベースに格納されます。

この情報は、コマンドラインから -start(または -restart)および -tcpip オプションを使用するか、またはアプリケーションの **StarTeam Server の構成** タブの **プロトコル** タブを使用して変更できます。

例:

```
starteamserver -start MyServer -tcpip 49201 starteamserver -start
MyServer -tcpip up
```

-u "DBUserName"

組み合わせるオプション: -new、-edit、-start、-P、-restart。

参照: -t、-dsn、-p。

サーバー構成がデータベースにアクセスするために使用するユーザー名を指定します。DBUserName に指定した値は starteam-server-configs.xml ファイルに格納されます。データベースのユーザー名は、コマンドラインから -view または -edit オプションを使用して確認または変更できます。必ず、このユーザー アカウントに対するパスワードも指定してください。変更内容は、次回にサーバー構成を起動したときに有効になります。starteamserver コマンドを使用して指定するユーザー名とパスワードが、データベースの有効なアカウントであることを確認してください。ユーザー アカウントがデータベースに見つからない場合、サーバー構成の起動は失敗します。

例:

```
starteamserver -edit MyServer -u SuperUser -p SuperUserPassword
```

-version

サーバーのバージョンおよびビルド番号を表示します。

例:

```
starteamserver -version
```

サーバーは次のようなメッセージを表示します。

```
StarTeam Server Version: x.x Build number: x.x.xxx
```

-view "ConfigurationName"

指定されたサーバー構成のセッションプロパティを一覧表示します。

例:

```
starteamserver -view StarDraw
```

関連参照

- コマンドライン操作

Checkout Trace のコマンドライン オプション

ここでは、Checkout Trace ユーティリティのコマンドライン オプションについて説明します。

一般に、デフォルトのオプションでコマンドラインからこのユーティリティを実行するには、次のようにします。
CheckoutTraceDump.exe -go Checkout Trace で使用できるオプションは、以下の表のとおりです。

オプション	説明
-go	このフラグを指定すると、デフォルトの設定でコマンドを実行します。
-path:<path>	バイナリ チェックアウト追跡(.cotrc)ファイルのフォルダ。 デフォルトは現在のフォルダ。
-outpath:<path>	出力(.csv)ファイルのフォルダ。 デフォルトは、バイナリ チェックアウト追跡(.cotrc)ファイルと同じフォルダ。
-file:<filespec>	入力として使用するバイナリ チェックアウト追跡ファイル。標準的なファイルシステムのワイルドカード(*、?)が使用できます。 デフォルトは"*.cotrc"(フォルダ内のすべてのチェクアウト追跡ファイル)。 このパラメータに複数のパスを指定することはできません。コマンド 1 つにつき、このパラメータは 1 つしか指定できません。
-ext:<extension>	チェックアウト追跡ファイルに使用するファイル拡張子。 この拡張子は、出力ダンプ ファイル名を作成するために、バイナリ チェックアウト追跡(.cotrc)ファイルの名前の末尾に追加されます。 デフォルトは、.csv。
-start:<start time>	関心のある最初の日時。 この日時以降に発生したチェックアウトだけが出力されます。 デフォルトでは、このユーティリティは日時によるフィルタは行いません。
-end:<end time>	関心のある最新日時を指定します。 この日時より前に発生したチェックアウトだけが出力されます。 デフォルトでは、このユーティリティは日時によるフィルタは行いません。
-project:<project name>	チェックアウト情報をフィルタするためのプロジェクト名。このプロジェクトからのチェックアウトだけが出力されます。デフォルトでは、このユーティリティはプロジェクトによるフィルタは行いません。すべてのプロジェクトが出力に含まれます。-project と -projectid の両方が指定されている場合は、-projectid が優先されます。
-projectid:<project ID>	チェックアウト情報をフィルタするためのプロジェクト ID。このプロジェクトからのチェックアウトだけが出力されます。デフォルトでは、このユーティリティはプロジェクト ID によるフィルタは行いません。すべてのプロジェクトが出力に含まれます。このオプションと -project オプションの両方が指定されている場合は、このオプションが優先されます。
-separator:<separator>	出力ファイル内の区切りに使用する文字列。デフォルトでは、"," が使われます。
-overwrite	このフラグを指定すると、既存のチェックアウト追跡ファイルが上書きされます。このフラグを指定しないと、ターゲット名の追跡ダンプファイルが既に存在する場合は、チェックアウト追跡ファイルはスキップされます。

VaultVerify のコマンドライン オプション

VaultVerify ユーティリティのコマンドライン オプションについて説明します。

一般に、コマンドラインから VaultVerify を実行するには、次のようにします: VaultVerify [options] "configuration"

デフォルトまたは与えられた -check オプションに基づいて、指定された StarTeam サーバー構成のデータ保管庫のアーカイブファイルに対して完全性チェックが実行されます。-repair オプションを指定した場合、VaultVerify は見つかった問題の修正を試みます。VaultVerify はサーバー構成のデータベースを開きますが、変更は行いません。VaultVerify で使用できるオプションは、以下の表のとおりです。

オプション	説明
-check {missing | corrupt | stray | all}	次のどの完全性チェックを実行するかを指定します。 missing: データベースと実際に存在するアーカイブファイルを比較し、不足しているファイルをチェックします。 corrupt: 既存のアーカイブ ファイルの完全性をチェックします(MD5、名前、フォルダ、.gz ファイル形式)。 stray: データベースに基づいて無関係なファイルをチェックします。サーバー構成が使用中の場合は、このオプションは使用できません。 all: すべての完全性チェックを実行します。 複数の -check オプションを指定することもできます。-repair オプションも参照してください。
-cf <folder path>	*破損ファイル用フォルダ* のパス名。このフォルダには、-repair を指定した場合に、corrupt チェックで見つかった問題のファイルが移されます。デフォルトの破損ファイル用フォルダは、C:¥Temp¥VVCorruptFiles です。
-dbhost <host>	指定された <configuration> 用のデータベースのホスト名を指定します。Windows 上では、-dbinstance も指定されている場合にのみ効力があります。Windows および Linux 上では、データベース サーバーがサーバーとは異なるホスト上で実行されている場合にのみ、このオプションを使用します。
-dbname <name>	指定された <configuration> 用のデータベース名を指定します。Windows 上では、このオプションは、-dbinstance も指定されている場合にのみ効力があります。これは、データベース名が ODBC DSN と異なる場合にのみ必要となります。Linux 上では、このオプションは、-dbinstance が指定されておらず、かつ Oracle サービス名が TNS 名と異なる場合にのみ使用されます。
-dbinstance <name>	このオプションは、Windows 上でのみ効力があります。使用すると、VaultVerify が構成ファイルで指定されている ODBC DSN を介する代わりに、直接データベースを開きます。SQL Server の場合、<name> はインスタンス名でなければなりません(例、「SSE2005_ST」)。デフォルト インスタンス名は、SQL Server の場合「MSSQLSERVER」で、SQL Server Express の場合は「SQLEXPRESS」である点に注意してください。Oracle の場合、サービス名でなければなりません(例、「ORCL」)。 データベース サーバーが異なるホスト上で実行されている場合、-dbinstance は -dbhost と共に使用しなければなりません。SQL Server の場合、データベースが ODBC DSN と異なる場合、さらに -dbname も使用される必要があります。Oracle の場合、-dbinstance が指定されていると -dbname は無視されます。
-dbpassword <password>	データベースのログオン パスワードを指定します。これを指定しないと、空のパスワードが使われます (構成内に保存されているパスワードは暗号化されているため、VaultVerify で使用することはできません)。Oracle を使って実行されているサーバー構成の場合、Oracle のパスワードは空であることはないので、このオプションを必ず指定しなければなりません。
-dbport <port>	データベース サーバーに接続するために使用する TCP/IP ポートを指定します。このパラメータは、ベンダーのデフォルトのデータベース ポート(Oracle の場合は

	1521）とは異なるポートを使用する場合に、Windows 以外のプラットフォームでのみ使用します。
-dbuser <user>	データベースに接続するために使用するユーザー ID を指定します。このパラメータを指定すると、StarTeam の <configuration> で指定されているユーザーよりも優先されます。このオプションを使用してよいのは、StarTeam テーブルを所有しているユーザーのみです。
-help（または -h、-?）	このコマンドの使用法を表示します。
-path <folder path>	starteam-server-configs.xml ファイルのフォルダ パスを指定します。このファイルが存在し、その中に指定の <configuration> が含まれていなければなりません。デフォルトでは、このファイルが現在の作業ディレクトリに見つからない場合、現在の作業ディレクトリの親フォルダにあるファイルがオープンされます。
-nosharereport	共有情報のレポート出力を抑止します。通常、各破損ファイルの共有パスがすべてレポートされます。このオプションは、共有パス情報を抑止し、アプリケーションの実行スピードを向上させ、レポート サイズの削減にも効果を表します。
-repair	アーカイブ ファイルの問題の修正を試みるかどうかを指定します。「破損した(Corrupt)」アーカイブは、「破損ファイル用フォルダ(corrupt file folder)」に移されます(-cf オプションを参照)。これらのファイルが有効なファイルリビジョンに対応する場合は、不明として扱われます。不明アーカイブの復元は、他のデータ保管庫ファイルから試みられます。-useca オプションが指定されている場合は、Cache Agent から復元が試みられます。「無関係な(Stray)」アーカイブは、「無関係ファイル用フォルダ(stray file folder)」に移されます(-sf オプションを参照)。 **注:** StarTeam の <configuration> が使用中の場合、-repair は無視されます。
-sf <folder path>	「無関係ファイル用フォルダ」のパス名。このフォルダには、-repair を指定した場合に、stray チェックで見つかった「無関係な」ファイルが移されます。デフォルトの「無関係ファイル用フォルダ」は、C:¥Temp¥VVStrayFiles です。
-t	検証の終了時に、経過時間を表示します。
-useca <host>:<port>	-repair を指定した場合、このオプションは、指定された MPX Cache Agent から不明ファイルの復元を試みます。Cache Agent は独立したキャッシュを維持するので、<host> および <port> は、リモート Cache Agent を指す必要があります。
-verbose	検証処理の進行に応じて、詳細なステータス情報を表示します。
“configuration”	構成名を指定します。VaultVerify に渡される構成名は、大文字小文字が区別されます。また、スペースが含まれている場合は、引用符で囲んで VaultVerify に渡す必要があります。

クライアントのコマンドライン操作

このセクションでは、クライアントのコマンドライン操作に関する参照情報を提供します。

このセクションの内容

サーバーのロックとロック解除： stcmd server-mode
サーバー構成をロックしたりロック解除したりするコマンドについて説明します。

ファイルの削除： stcmd remove
バージョン管理からファイルを削除するコマンドについて説明します。

個人用オプションの設定： stcmd set-personal-options
個人用オプションの設定や表示を行うコマンドについて説明します。

ファイル ステータスの更新： stcmd update-status
ファイルのステータスを更新するコマンドについて説明します。

共通オプション

一部のオプションは、すべてのコマンドまたはほとんどのコマンドで使用できます。これらのオプションはどのコマンドでも同じ意味を持つため、ここでまとめて説明し、あとで繰り返すことはしません。必ずしもすべてのコマンドに使える訳ではないオプションや、コマンドごとに意味が異なるオプションについては、そのコマンドと一緒に説明します。

すべてのコマンドライン構文は、Windows 構文ではなく UNIX 構文として解釈されます。

すべてのオプションは、使用しているオペレーティング システムでの規則に従って、ハイフン(-)またはスラッシュ(/)で記述します。ヘルプの記述でオプションに引用符が必要とある場合でも、使用しているオペレーティング システムによっては省略しても受け付けられることがありますが、引用符は常に使用することをお勧めします。そうしておけば、統一がとれるうえ安全です。

Windows オペレーティング システムでは、オプションの一部に空白がある場合に引用符で囲む必要があります。このため、リビジョンのコメントが複数の単語を含む場合は引用符で囲む必要があります。1 つの単語から成るコメントは引用符で囲む必要はありませんが、引用符を使用してもかまいません。Java コマンドライン経由で送られたコマンドは、作業フォルダおよびファイル名に空白が含まれていると失敗します。UNIX の名前は、二重引用符で囲まない限り空白を含むことはできません。

構文表記規則

コマンドラインの構文では、以下の表記規則を使用します:

[ ]

省略可能な構文を囲みます。

|

同時に選択することのできない選択肢を区切ります。選択肢の 1 つだけを選択します。

-?

コマンドの構文と各オプションの説明を表示します。

-? は、ヘルプも起動します。-help および -h は、-? と同じ意味です。-? は構文には指定されていませんが各コマンドで使用できます。この情報は stdout ではなく stderr に送られます。Windows コマンド プロンプトから stderr の情報を取得するには、「2>」を使用してください(stdout の情報を取得する「>」ではなく)。

-active

アクティブな処理アイテムを示します。

-cmp

ワークステーションとサーバー間で送られるすべてのデータを圧縮し、到着時に解凍します。このオプションを省略すると圧縮は行われません。

圧縮は、クライアントとサーバーが低速の接続を介して通信する場合に便利です。圧縮を使用するかどうかを決定するために、小さいテストケースで試してみることをお勧めします。データの圧縮と解凍に費やされる時間が発生しても、低速接続を介して長時間かけて非圧縮データを送るより良いかどうかという観点で検討しなければなりません。

-csf

コマンドの実行によって、-p オプションで指定したフォルダを基本となる StarTeam のフォルダにマップする場合に、-csf を指定すると、StarTeam のフォルダの名前の大文字小文字が区別されるようになります。このオプションは、フォルダ内のファイル名の大文字小文字の扱いには適用されません。たとえば、-csf を指定すると、「doc」という名前の StarTeam フォル

ダと「Doc」という名前の StarTeam フォルダは別フォルダと見なされます。このオプションを指定しないと、どちらのフォルダも同じ「doc」フォルダと見なされます。

デフォルトでは、StarTeam フォルダの名前は大文字小文字が区別されません。

-csf を使用してもしなくてもファイル名があいまいな場合はエラーとなります。たとえば、-csf を使用する際、Doc と doc フォルダの両方がある場合 2 つのフォルダの名前はあいまいです。-csf を使用しない場合、スペルが同じフォルダ名はあいまいです。

-encrypt

ワークステーションとサーバー間で送られるすべてのデータを暗号化し、到着時に復号します。このオプションを省略すると暗号化は行われません。暗号化すれば、安全が確保されていないネットワーク回線上で、ファイルなどのプロジェクト情報が第三者に不正に読み取られないようにすることができます。

完全な構文は以下のとおりです。

```
-encrypt encryptionType
```

暗号化レベルには以下のものがあります。

RC4: RSA RC4 ストリーム暗号(高速)

RC2_ECB: RSA RC2 ブロック暗号(Electronic Codebook)

RC2_CBC: RSA RC2 ブロック暗号(Cipher Block Chaining)

RC2_CFB: (Windows プラットフォームのみ)RSA RC2 ブロック暗号(Cipher Feedback)

これらの暗号化レベルは高速なものから順番に並べられています。低速の暗号化レベルほど、安全性は高くなります。

注: Windows 以外のプラットフォームでは、暗号化処理で使用される公開鍵と秘密鍵が自動的には生成されません。公開鍵と秘密鍵は、ユーザーのホームディレクトリ内のファイルに保存されます。このオプションのファイルの名前は、.starteam です。この中に keyfile という変数またはシェル変数があります。keyfile 変数は、公開鍵および秘密鍵を含むファイルの場所を指定します。keyfile 変数を指定しないとエラーとなります。keyfile 変数を指定したがファイルが存在しない場合、StarTeam クライアントはランダムな鍵のペアを生成し、ファイルを作成し、その中に鍵を格納します。必ずこのファイルの安全を確保してください。たとえば、UNIX では所有者しか読み取れないようにしておきます。

-eol [on|off]

テキストファイルの行末変換を実行するかどうかを指定します。

-epwdfile

ユーザーのパスワードを、ローカル ファイルに暗号化された値として保存します。この機能は、自動ビルド スクリプトをサポートしており、人手を介さずに実行しなければなりません。このスクリプトでは、特定のユーザー名と、そのユーザー名に対する暗号化されたパスワードを保存したファイルの名前で stcmd を呼び出します。すると、そのパスワードが内部的に解読され、ネットワーク上を平文テキストとして転送されることなく、サーバーに渡されます。

-epwdfile キーワードでは、暗号化したパスワードを保存したファイルへのパスを指定します。-pwdfile と同様、-epwdfile は -p オプションの一部に指定されているパスワードを上書きします。これにより、他者がコマンドラインに指定されたユーザーのパスワードを見てしまうことを回避することができます。完全な構文は以下のとおりです。

```
-epwdfile  “filePath”
```

次の表は、暗号化パスワードを保管するために使用することのできる、コマンドの構文を示したものです。

アクション	構文
暗号化されファイルに保存されるパスワードを入力するよう、プロンプトを表示される場合。	`stcmd store-password -epwdfile "filePath"`
暗号化パスワードを、コマンドラインに平文テキストとして含める場合。(このアクションでは、この平文テキストの値でネットワーク転送を行うわけではない点に留意してください)。	`stcmd store-password -epwdfile "filePath" -password "password"`

一旦、暗号化されたパスワードが保存されたら、他の stcmd コマンドにパラメータとして「-epwdfile "filePath"」を指定できます。たとえば、

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

-f NCI

チェックインが必要なすべてのファイルにコマンドを適用します。

-filter と同時に指定した場合は、-f NCI は無視されます。

files...

コマンドで使用するファイルを名前またはファイル名パターン("*.c" など)で指定します。UNIX の構文規則の方が Windows よりも具体的なので、すべてのオプションが UNIX の構文規則で解釈されます。このため、「すべてのファイル」を指定するには、"*"」ではなく、"*.*" を使用します。パターン "*.*" は、「名前と拡張子を持つすべてのファイル」を意味します。たとえば、"star*.*" は、starteam.doc や starteam.cpp と一致しますが、starteam とは一致しません。これらすべてを検索するには、"star*" を使用します。

このオプションを省略するとデフォルトは "*" です。このオプションは、常に最後に指定する必要があります。このオプションより後ろのオプションはすべて無視されます。

すべてのファイルを示すのに "*" ではなく * を使用すると、UNIX シェルはそれを一連のアイテムに拡張してそれをオプションのグループとして stcmd コマンドに渡します。たとえば、作業ファイルなしのファイルをチェックアウトしている場合、これにより問題が生じる可能性があります。混乱を避けるために、"*" を使用することをお勧めします。

複数のファイル パターンを使用する場合、各パターンをそれぞれ引用符で囲む必要があります。たとえば、"*.bat" "*.c" は使用できますが "*.bat *.c" は使用できません。

注: このオプションは、プラットフォームに関係なく必ず引用符で囲むことをお勧めします。ただし、その理由はプラットフォームごとに異なります。Windows プラットフォームでは、明示的に引用符で囲まない限り、空白を含むファイル名とフォルダ名は正しく解釈されません。UNIX プラットフォームでは、引用符を使用しない場合、シェルがオプションを展開し、展開して作られたアイテムのリストがクライアントに渡されます。しばしば、これによって予期しない結果が生じます。このような結果を避けるために、このオプションは引用符で囲むことをお勧めします。UNIX シェルによるオプションの展開が必須の場合にのみ、引用符を省略してください。いずれのプラットフォームでも、バッチ スクリプトが原因不明で失敗する場合、引用符の省略が原因となっている可能性があります。

ファイルの指定では、以下の特殊文字を使用できます。

*

任意の文字列(空白の文字列を含む)と一致します。例えば、* は、拡張子の有無に関係なくあらゆるファイル名と一致します。"xyz*" は、「xyz」、「xyz.cpp」、「xyzutyfj」のいずれとも一致します。

-?

任意の 1 文字と一致します。例えば、"a?c" は、「abc」とは一致しますが、「ac」とは一致しません。

[...]

左右のかぎ括弧に囲まれた文字のいずれかと一致します。

2 つの文字の間にハイフン(-)を入れると、その範囲内のいずれかの文字と一致します。

左かぎ括弧([)の直後に感嘆符(!) またはキャラット(^)を置くと、括弧内の文字とは一致しません。カギ括弧で囲まれていない任意の文字と一致します。例えば、"x[a-d]y" は、「xby」 とは一致しますが、「xey」とは一致しません。一方、"x[!a-d]y" は、「xey」とは一致しますが、「xby」とは一致しません。

ハイフン(-)または右カギ括弧( ])そのものと一致させるには、カギ括弧内の最初または最後の文字として含めます。

パターン内でアスタリスク(*)、疑問符(?)、または左カギ括弧([)そのものと一致させるには、その直前にエスケープ文字(バックスラッシュ ¥)を入れる必要があります。

-filter

ファイルステータスフィルタを設定します。指定できるステータスは、C = 最新、M = 変更済み、O = 古いリビジョン、N = ビュー外、I = 作業ファイルなし、G = マージ、U = 不明、です。例えば、CM と指定すると、ステータスが「最新」または「変更済み」のファイルのみにコマンドが適用されます。

-filter は、-f NCI よりも優先されます。G、O、または、U を使用する場合は、-i または -o も指定する必要があります。そうでなければ、G、O、または U の指定は無視されます。

-filter は、-f NCO よりも優先されます。G、M、O、U のいずれかを指定する場合は、-merge または -o も指定して、チェックアウト処理を強制しなければなりません。これを行わないと、G、M、O、または U の指定は無視されます。

-fp

指定されたフォルダの作業フォルダ(作業ディレクトリ)をオーバライドします。これは、フォルダの代替作業パスを設定するのと同じ結果となります。

このオプションでは、StarTeam ビューで指定されたものとは別の作業フォルダを使用することができますが、もっとも重要な役割はプラットフォーム間の互換性を提供することです。たとえば、UNIX と Windows システムでは、ドライブとディレクトリパス名の指定方法に互換性がありません。

パス D:¥MYPRODUCT¥DEVELOPMENT¥SOURCE は Windows プラットフォームでは有効ですが UNIX プラットフォームでは無効です。自分のプラットフォームが StarTeam プロジェクトで指定されたパスを解釈できない場合、このオプションを使用して作業パスを定義します。

引用符の前に置かれた円記号(¥)は、エスケープ文字として解釈されます。したがって、次の例はエラーとなります。

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

これは次のように解釈されます。

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

これを避けるため、「C:¥」の最後の文字を次のようにエスケープします。

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

または、-rp パスが「C:¥orion¥」のようにルート フォルダで終わらないパスの場合は、次のようにしてもこの問題を回避できます。

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

完全な構文は以下のとおりです。

```
-rp "folderName"
```

フォルダは Windows の用語で、StarTeam のユーザー インターフェイスに表示されます。UNIX プラットフォームでは、ディレクトリが正しい用語です。

-h

ヘルプを表示します。-help は構文には示されませんが、どのコマンドでも使用できます。

-help

ヘルプを表示します。-help は構文には示されませんが、どのコマンドでも使用できます。

-i

ファイルステータスが「マージ」、「古いリビジョン」、「不明」のいずれかの場合に、ユーザーにチェックインするかどうかの確認を求めます。

-is

コマンドをすべての子フォルダに適用します。このオプションを省略すると、コマンドは指定したフォルダにのみ適用されます。

add-folder でこのオプションを使用すると、フォルダの分岐全体を StarTeam フォルダ階層に追加できます。また、add-project で使用すると、ルート フォルダだけではないプロジェクトを作成できます。

-l

ファイルをロックします。

-mark

変更要求には「解決」、要件には「完了」、タスクには「終了」のマークを付けます。

-nel

ファイルに対して非排他的ロックをします。

-nologo

著作権情報が表示されないようにします。-nologo は構文には示されませんが、どのコマンドでも使用できます。

–nomove

既に付加されているラベルは移動しません。

–o

強制的にチェックインします。

–p

使用するビューやフォルダ、およびサーバーにアクセスするために必要なユーザー名とパスワードを指定します。完全な構文は以下のとおりです。

```
-p "userName:password@hostName:endpoint/projectName/[viewName/][folderHierarchy/]"
```

例:

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

- ユーザー名を省略すると、現在のユーザー名が使われます。
- パスワード を省略すると、パスワードを入力するよう要求されます。ユーザーがパスワードを入力するとき、文字は画面上に表示されません。
- ホスト名を省略すると、デフォルトで localhost となります。
- エンドポイント(ポート番号)は、必ず入力する必要があります。デフォルトは 1024 です。
- プロジェクト名は必須です。
- ビューの指定にはビュー階層を使用します。ビュー名の区切り文字にはコロン(:)を使用します。ビュー階層には、必ずルート ビューを含めます。たとえば、「StarDraw:Release 4:Service Packs」は、StarDraw ルート ビューの孫で Release 4 ビューの子の Service Packs ビューを使用することを意味します。ビュー名を省略すると、ルート ビューが使用されます。そのビューがプロジェクト内で唯一の名前を持つビューの場合は、ビュー名だけを指定することもできます。ただし、後から同じ名前のビューが作られる可能性もあるのでこの方法はお勧めできません。
- フォルダの指定にはフォルダ階層を使用します。フォルダ名の区切り文字にはスラッシュ(/)を使用します。フォルダ階層には、ルート フォルダを含めてはなりません。ファイルがビューのルート フォルダにある場合はフォルダ階層を省略します。たとえば、ビューのルート フォルダが StarDraw でファイルの階層が「StarDraw/SourceCode/Client」ならば、単に「SourceCode/Client」を使用します。

このオプションで使用する変数に区切り文字として使われている文字が含まれている場合、その文字にはパーセント記号(%)で始まる 16 進数コードを使用します。たとえば、パスワードの中に「@」が表れる場合、「%40」で置き換える必要があります。

":" には、"%3a" を使用

"/" には、"%2f" を使用

"@" には、"%40" を使用

"%" には、"%25" を使用

UNIX および他のオペレーティング システムでは、一部の特殊文字の先頭に円記号 "¥" または別のエスケープ文字を指定する必要があります。-p オプションでは、そのような文字を 16 進数コードで置き換えることができます。たとえば、UNIX で "¥<" とする代わりに "%3c" を使用することができます。

空白には「%20」を使用します。

「<」には、「%3c」を使用します。

「>」には、「%3e」を使用。

-pwdfile

ユーザーのパスワードを保存したファイルへのパスを指定します。このオプションは -p オプションで使われるパスワードに優先します。これによって、コマンドライン上で他人にパスワードを見られるのを防止できます。完全な構文は以下のとおりです。

```
-pwdfile "filePath"
```

-q

処理状況レポートを表示しません。このオプションを省略すると、アクションを実行するたびに各アクションに関するメッセージが画面に表示されます。

-r

チェックインの理由を示します。

-rf

チェックインの理由が記述されたファイルの名前を指定します。

-ro

処理の完了後にファイルを読み取り専用にします。

-rp

StarTeam ビューのルート フォルダの作業フォルダ(作業ディレクトリ)を指定またはオーバーライドします。stcmd add-project コマンドは、このオプションを使用して新規ビューのルート フォルダの作業フォルダを指定します。他のコマンドは、既存の作業フォルダを置き換えるために使用します。

このオプションでは、StarTeam ビューで指定されたものとは別の作業フォルダを使用することができますが、もっとも重要な役割はプラットフォーム間の互換性を提供することです。たとえば、UNIX と Windows システムでは、ドライブとディレクトリパス名の指定方法に互換性がありません。

パス D:¥MYPRODUCT¥DEVELOPMENT¥SOURCE は Windows プラットフォームでは有効ですが UNIX プラットフォームでは無効です。自分のプラットフォームが StarTeam プロジェクトで指定されたパスを解釈できない場合、このオプションを使用して作業パスを定義します。

UNIX シェルでは、引用符など特定の文字列の前に置かれた円記号(¥)は、エスケープ文字として解釈されます。したがって、次の例はエラーとなります。

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

これは次のように解釈されます。

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

これを避けるため、「C:¥」の最後の文字を次のようにエスケープします。

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

または、-rp パスが 「C:¥orion¥」のようにルート フォルダで終わらないパスの場合は、次のようにしてもこの問題を回避できます。

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

完全な構文は以下のとおりです。

```
-rp "folderName"
```

フォルダは Windows の用語で、StarTeam のユーザー インターフェイスに表示されます。UNIX プラットフォームでは、ディレクトリが正しい用語です。

-rw

処理の完了後にファイルを読み取り/書き込み可能にします。

-stop

しばしば -x と一緒に使われます。最初のエラーが発生した時点でコマンドラインの実行を停止します。このオプションを省略するとエラーが発生しても実行が継続します。

-u

ファイルのロックを解除します。

-v

バージョンラベルを指定します。

-x

対話モードとバッチモードを切り替えます。このオプションを指定しないと、エラー メッセージを対話式に確認しなければならなくなり、終了コードが返されない場合があります。このオプションを使用すると、エラー メッセージは表示されず、終了コードが設定されます。終了コードは処理に成功した場合は 0、失敗した場合は 1 です。

関連概念

特殊文字

関連参照

クライアントのコマンドライン操作

特殊文字

*: 任意の文字列(空白の文字列を含む)と一致します。たとえば、"x*z" は、「xyz」にも「xz」にも一致します。?: 任意の 1 文字と一致します。たとえば、"a?c" は、「abc」とは一致しますが、「ac」とは一致しません。

[...]: 左右のかぎ括弧に囲まれた文字のいずれかと一致します。

ハイフン(-)で区切られた文字のペアは照合する文字の範囲を指定します。左かぎ括弧([)の直後に感嘆符(!)またはキャラット(^)を置くと、括弧内の文字は一致せず、カギ括弧で囲まれていない任意の文字と一致します。ハイフン(-)または右括弧(])は、かぎ括弧内の最初の文字または最後の文字とすることで、検索されるようになります。たとえば、"x[a - d]y" は「xby」と一致しますが、「xey」とは一致せず、"x[!a - d]y" は「xey」と一致しますが「xby」とは一致しません。パターン内でアスタリスク(*)、疑問符(?)、または左かぎ括弧([)をそのまま使用したい場合は、その直前にエスケープ文字(¥)を入れる必要があります。

すべてのファイルを示すため "*" の代わりに * を使用すると、UNIX シェルはそれを一連のアイテムに拡張してそれをオプションのグループとして stcmd コマンドに渡します。これにより問題が生じる可能性があるので(たとえば、不足ファイルのチェックアウトを行う場合)、"*" を使用して不要な混乱を避けることをお勧めします。複数のファイル パターンを使用する場合、各パターンをそれぞれ引用符で囲む必要があります。たとえば、"*.bat" "*.c" は使用できますが "*.bat *.c" は使用できません。

これらの特殊文字は、一部のコマンドで使用できる files... オプションにも適用されます。

関連参照

クライアントのコマンドライン操作

終了コード

stcmd コマンドは、コマンドで -x オプションを使用していると終了コードを返します。終了コードには以下のものがあります。処理に成功した場合は 0、失敗した場合は 1 です。少なくとも 1 つの指定ファイル パターンが一致しなければ 101、いずれの指定ファイル パターンも一致しなければ 102 です。stcmd diff コマンドには、終了コードを返すもう 1 つのオプション(-e)があります。-e オプションには上記と異なる意味を持つ 3 つの終了コード(0、1、2)があります。また、一緒に -x も使用しなければ 1 が返されない可能性があります。

Windows プラットフォームの場合

バッチ ファイル内で ERRORLEVEL を使用してコマンドの結果に基づいて処理を実行することができます。たとえば、バッチ ファイル内で stcmd コマンドの後に以下のステートメントを使用できます。

```
IF ERRORLEVEL int statement
```

ここで、int は、0 または 1 です。

例:

```
IF NOT ERRORLEVEL 1 GOTO OPOK
ECHO ERROR OCCURRED AT STEP5>LOGFILE.TXT.

:OPOK
```

疑似環境変数 %ERRORLEVEL% を使用することもできます。たとえば、シェルまたはコマンドラインで以下を使用することができます(stcmd コマンドの後)。

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

UNIX の場合

各シェルが独自に終了コードを評価する方法を持っています。たとえば、Bourne シェルでは stcmd コマンドの後に以下のステートメントを入れることができます。

```
if [ return ]; then statement
```

関連参照

クライアントのコマンドライン操作

ファイルの追加: stcmd add

stcmd add は、コマンドラインからプロジェクトにファイルを追加するために使用します。

同時に、追加したファイルを処理アイテムにリンクすることができます。このコマンドを使用して正常に追加されたファイルは、すべて処理アイテムのチップ リビジョンにリンクされ固定されます。-active オプションを使用して現在アクティブな処理アイテム(以前に、ユーザーのワークステーションで StarTeam クライアントを使用して設定された)を指定します。

アクティブなアイテムがないか、または別なアイテムを使用する場合、処理アイテムのタイプを示すオプション(-cr、-req、-task)に続いてプロジェクト ビューのルート フォルダからアイテムまでの完全なパスを使用します。この際、フォルダ名間の区切りにはスラッシュ(/)を使用します。ビュー外の処理アイテムについては、完全なフォルダ パスの前にプロジェクト名とビュー名を指定します。ビュー パスは、コロン(:)で区切ります。たとえば、 -cr MyProject/RootView:ChildView/SourceCode/37 は、MyProject プロジェクトの ChildView ビューの SourceCode フォルダにある変更要求 37 を指定します。実行中、処理ではまず処理アイテムが現在のビューにあるものとみなし、現在のビューを確認してフル パスがそのビュー内のフォルダ パスに対応するかどうか判断します。処理アイテムが現在のビューにない場合、処理外アイテムとして扱われ、プロジェクトとビューから処理アイテムの検索を開始します。

同時に、-mark オプションを使用して処理アイテムをそのタイプに従って「解決」、「終了」、「完了」とマークすることもできます。たとえば、変更要求は解決とマークできます。アイテムは、すべてのファイルが正常に追加されなかった場合は、「解決」、「終了」、「完了」とマークされません。

構文

このコマンドの構文は次のとおりです。

```
stcmd add -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-csf] [-encrypt encryptionType]
[-is] [-q] [-x] [-stop] [-rp "folderPath" | -fp "folderPath"] [- l | -u | -nel] [-ro | -rw]] [-d
"description"]
[-vl "labelName"] [-eol [on | off]] [[ -active | [-cr | -req | -task ] processItemPath] [-mark]] [-short]
[files...]
```

オプション	説明
-active	アクティブな処理アイテムを指定します。
-cr -req -task	プロジェクト ビューのルート フォルダから処理アイテムとして使用する変更要求、要件、またはタスク番号への完全なパスを指定します。フォルダ名の区切り文字にはスラッシュ(/)を使用します。 ビュー外の処理アイテムについては、完全なフォルダ パスの前にプロジェクト名とビュー名を指定します。ビュー パスは、コロン(:)で区切ります。たとえば、 -cr MyProject/RootView:ChildView/SourceCode/37 は、MyProject プロジェクトの ChildView ビューの SourceCode フォルダにある変更要求 37 を指定します。 実行中、処理ではまず処理アイテムが現在のビューにあるものとみなし、現在のビューを確認してフル パスがそのビュー内のフォルダ パスに対応するかどうか判断します。処理アイテムが現在のビューにない場合、処理外アイテムとして扱われ、プロジェクトとビューから処理アイテムの検索を開始します。
-d	新規に追加するファイルの説明を、指定した文章に設定します。この文章は二重引用符で囲みます。
-eol	作業中のテキスト ファイルをサーバーのリポジトリに転送する際に、行末マーカー(EOL)を復帰/改行(CR/LF)に自動変換します。このオプションが on の場合、作業ファイルの行末マーカー(EOL)を CR/LF に変換します。デフォルトでは、このオプションは off になっており、行末変換は行われません。 例えば、UNIX で稼働しているコンピュータから作業ファイルをリポジトリに追加する際に、テキストファイルを CR/LF として保存したい場合は、このオプションを on にします。
-l	ビューに追加したファイルをすべてロックします。-l、-u、または -nel を指定しない場合、デフォルトではファイルはロックされません。
-mark	すべてのファイルが正常に追加された場合に、処理アイテムのステータスを「解決」(変更要求の場合)、「終了」(タスクの場合)、「完了」(要件の場合)に変更します。ファイルは、新しいステータスを持つリビジョンに固定されます。
-nel	追加したファイルすべてに非排他的ロックをします。

-ro	この操作の後、作業ファイルを読み取り専用にします。このオプションを指定しないと、ファイルは操作の前と同じ状態のままとなります。通常は、 -ro を使用して、自分がロックしていないファイルを自分自身が編集することを防ぎます。 -ro は、-l、-u、または -nel を一緒に使用しなければなりません。-ro を使用する場合は、-rw は使用できません。
-rw	この操作の後、作業ファイルを読み取り/書き込み可能にします。このオプションを指定しないと、ファイルは操作の前と同じ状態のままとなります。 -rw は、-l、-u、または -nel を一緒に使用しなければなりません。-rw を使用する場合は、-ro は使用できません。
-u	新規に追加したファイルをロック解除にします。
-vl	新規ファイルに適用するラベルを指定します。ラベルは二重引用符で囲みます。このオプションは、コマンド内で複数回指定できます。ここで指定できるラベルはビュー ラベルまたはリビジョン ラベルです。ただし、アプリケーション内に既に存在するラベルでなければなりません。

例

以下の例は、stcmd add を使用してステータスが「ビュー外」のすべての .doc ファイルをルート フォルダ StarDraw（StarDraw プロジェクトの StarDraw ビュー内）の子である User Manual に追加します。全ファイルをロックし、説明「First daft of chapter」を追加します。

```
stcmd add -p "JMarsh:password@Orion:1024/StarDraw/StarDraw/User Manual" -l-d "First draft of chapter" "*.doc"
```

関連参照

クライアントのコマンドライン操作

フォルダの追加: stcmd add-folder

stcmd add-folder は、コマンドラインからビューにフォルダを追加するために使用します。フォルダは、ルート フォルダまたはそのビュー内の任意の他のフォルダに追加することができます。新規 StarTeam フォルダの作業フォルダは、デフォルトでユーザーのワークステーションではなく StarTeam 内に作られます。作業フォルダの名前は StarTeam フォルダと同じです。これは、StarTeam フォルダの親の作業フォルダの子フォルダです。

たとえば、「Wizard」という名前の StarTeam フォルダを作成するとします。Wizard は、作業フォルダが「C:¥StarDraw」である StarTeam フォルダの子です。したがって、Wizard の作業フォルダは「C:¥StarDraw¥Wizard」となります。

-is オプションを使用すると、プロジェクト ビューのフォルダ階層にフォルダの分岐を追加することができます。-is を使用する場合、-rp または -fp を使用して子フォルダが新規 StarTeam フォルダの子フォルダとなるフォルダをユーザーのワークステーションで指定します。その際には、子フォルダの親へのパスを直接指定できる -fp の使用を推奨します。これに対してビューのルート フォルダに使用されている作業フォルダへのパスを指定する -rp は、ルート フォルダから新規フォルダへの階層内の StarTeam フォルダ名をユーザーが指定するパスに追加します。-rp および -fp は、-is オプションを使用した場合にのみこのコマンドに対して有効です。

構文

このコマンドの構文は次のとおりです。

```
stcmd add-folder -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-csf] [-encrypt encryptionType]
[-is] [-q] [-x] [-stop] [-rp "folderPath" | -fp "folderPath"] -name "folderName"
[-d "description"] [-ex "excludeType"] [-exlist "fileMask" | -exfile "fileName"]
```

オプション	説明
-d	フォルダに対する説明を指定します。最大 254 文字まで入力できます。
-ex	新規フォルダで使用する除外リストを指定します。除外リストによって、特定のファイルや特定の種類のファイルが表されなくなります。フォルダの作業フォルダ内の作業ファイルが「ビュー外」ステータスであっても、除外リストの指定と一致する場合は、その作業ファイルは表示されません。それは、あたかも存在しないかのように扱われます。 例えば、ファイルを保存するたびにそのファイルの自動バックアップ コピー(.bak ファイル)を作成するアプリケーションでファイルを作成しているとしましょう。作業フォルダには .bak ファイルがいくつか入っているかもしれませんが、それらをチェックイン(あるいはチェックアウト)する必要がないとします。アプリケーションから、このような .bak ファイルが見えるのが煩わしい場合は、それらを除外します。ファイルの除外は、フォルダ単位で行われます。ただし、除外リストを親フォルダから継承することができます。 完全な構文は以下のとおりです: -ex excludeType 除外タイプには以下のものがあります。 inherit はこのフォルダが親フォルダで使われている除外リストを継承し、-exfile または -exlist で指定された除外リストを使用するように指定します(リストが作成されている場合)。これがデフォルトです。 local は、このフォルダが、-exfile または -exlist で指定された除外リストのみを使用するように指定します。 none は、このフォルダが、-exfile または -exlist が指定されていても除外リストを全く使用しないように指定します。
-exfile	このフォルダに対するローカルな除外リストを含むファイルへのパスを指定します。除外リストの内容の説明については、-exlist を参照してください。
-exlist	このフォルダに対するローカルな除外リストを指定します。最大 254 文字まで入力できます。ファイル指定(標準的な * と ? のワイルドカードを使用)は、カンマ、スペース、またはセミコロンで区切って複数入力できます。カンマ、空白、またはセミコロンをファイル指定の一部に使用する場合は、二重引用符で指定を囲みます。 例: *.exe,*.dll p*z.doc;*.t?t "test *.*" 除外リスト内で二重引用符を使用している場合や、除外リストが長い場合は、-exfile オプションの使用をお勧めします。-exlist では、除外リスト内の引用符の前に、システムやシェルに応じたエスケープ文字を付ける必要があります。例えば、NT システムの場合はキャラット(^)を付けます。-exfile を使用すると、エスケープ文字を使う必要がありません。

-name	フォルダに対する名前を指定します。最大 254 文字まで入力できます。ファイル内では、除外リストに二重引用符が含まれていても、エスケープ文字は不要です。

例

次の例は、stcmd add-folder を使用して、Wizard という名前のフォルダを StarDraw プロジェクト ビューのルート フォルダである StarDraw フォルダの子フォルダとして作成します。さらに、Wizard のローカル除外リストも設定します。デフォルトで Wizard はその親フォルダの除外リストを継承し、ローカルの除外リストも使用します。

```
stcmd add-folder -p "JMarsh:password@Orion:1024/StarDraw/StarDraw/" -name "Wizard" -d "StarDraw setup wizard"
-exlist "*.bak"
```

次の例は、前の例と同じフォルダを作成します。ただし、今度は子フォルダを含みます。この場合、パスが「C:¥Wizard」のフォルダには子フォルダ（Source、Spec、および Doc）があり、それらはすべて Wizard に加えてアプリケーション フォルダとして追加されます。すべての新規フォルダ（Wizard、Source、Spec、および Doc）は、-fp の設定にかかわらずサーバーによりデフォルトの作業フォルダが自動的に割り当てられます。Wizard は Source、Spec、および Doc の親となります。StarDraw は Wizard の親です。

```
stcmd add p "JMarsh:password@Orion:1024/StarDraw/ StarDraw/" -name "Wizard" -d "StarDraw setup wizard" -is -fp
"C:¥Wizard" exlist "*.bak"
```

関連参照

クライアントのコマンドライン操作

プロジェクトの追加: stcmd add-project

stcmd add-project は、コマンドラインからサーバー構成にプロジェクトを追加するために使用します。プロジェクトを作成すると、そのルート ビューとルート ビューのルート フォルダも作成されます。このコマンドでは、-rp オプションはそのルート フォルダの作業フォルダを指定します。

-is を使用すると作業フォルダの子フォルダを StarTeam のフォルダ階層でルート フォルダの子フォルダとして使用することができます。

構文

このコマンドの構文は次のとおりです。

```
stcmd add-project [-pwdfile "filePath" ] [-cmp] [-encrypt encryptionType] [-is] [-q] [-x]
[-stop] -s "serverName" -name "projectName" -rp "folderPath" [-d "description"]
[-kw "fileMask" |-kwfile "fileName"] [-ex "excludeType" ] [-exlist "fileMask" |-exfile "fileName"]
```

オプション	説明
-d	プロジェクトに対する説明を指定します。最大 254 文字まで入力できます。
-ex	プロジェクトのルート フォルダで使用する除外リストを指定します。 除外リストによって、特定のファイルや特定の種類のファイルが表されなくなります。フォルダの作業フォルダ内の作業ファイルが「ビュー外」ステータスであっても、除外リストの指定と一致する場合は、その作業ファイルは表示されません。それは、あたかも存在しないかのように扱われます。 例えば、ファイルを保存するたびにそのファイルの自動バックアップ コピー(.bak ファイル)を作成するアプリケーションでファイルを作成しているとしましょう。作業フォルダには .bak ファイルがいくつか入っているかもしれませんが、それらをチェックイン(あるいはチェックアウト)する必要がないとします。アプリケーションから、このような .bak ファイルが見えるのが煩わしい場合は、それらを除外します。ファイルの除外は、フォルダ単位で行われます。ただし、除外リストを親フォルダから継承することができます。完全な構文は以下のとおりです。 -ex excludeType 除外タイプには以下のものがあります。 inherit はルート フォルダが親フォルダで使われている除外リストを継承し、-exfile または -exlist で指定された除外リストを使用するように指定します(リストが作成されている場合)。ルート フォルダが何も継承しないとしても、これがデフォルトです。 local は、ルート フォルダが、-exfile または -exlist で指定された除外リストのみを使用するように指定します。 none は、ルート フォルダが、-exfile または -exlist が指定されていても除外リストを全く使用しないように指定します。
-exfile	ルート フォルダに対するローカルな除外リストを含むファイルへのパスを指定します。除外リストの内容の説明については、-exlist を参照してください。
-exlist	ルート フォルダに対するローカルな除外リストを指定します。最大 254 文字まで入力できます。ファイル指定(標準的な * と ? のワイルドカードを使用)は、カンマ、スペース、またはセミコロンで区切って複数入力できます。カンマ、空白、またはセミコロンをファイル指定の一部に使用する場合は、二重引用符で指定を囲みます。例: *.exe,*.dll p*z.doc;*.t?t "test *.*" 除外リスト内で二重引用符を使用している場合や、除外リストが長い場合は、-exfile オプションの使用をお勧めします。-exlist では、除外リスト内の引用符の前に、システムやシェルに応じたエスケープ文字を付ける必要があります。例えば、NT システムの場合はキャラット(^)を付けます。-exfile を使用すると、エスケープ文字を使う必要がありません。
-kw	キーワードを使用するファイルの拡張子を指定します。最大 254 文字まで入力できます。ファイル指定(標準的な * と ? のワイルドカードを使用)は、カンマ、スペース、またはセミコロンで区切って複数入力できます。カンマ、空白、またはセミコロンをファイル指定の一部に使用する場合は、二重引用符で指定を囲みます。例: *.cpp,*.h p*z.doc;*.t?t "test *.*"

	キーワードリスト内で二重引用符を使用している場合や、キーワードリストが長い場合は、-kwfile オプションの使用をお勧めします。-kwlist では、キーワードリスト内の引用符の前に、システムやシェルに応じたエスケープ文字を付ける必要があります。例えば、NT システムの場合はキャラット(^)を付けます。-kwfile を使用すると、エスケープ文字を使う必要がありません。
-kwfile	キーワードを使用するファイル拡張子を持つファイルへのパスを指定します。
-name	プロジェクトの名前を指定します。最大 254 文字まで入力できます。
-s	サーバーを指定します。完全な構文は以下のとおりです。 -s "userName:password@host:portNumber" 例: -s "JMarsh:password@orion:49201" ユーザー名を省略すると、現在のユーザー名が使われます。前記の例では、ユーザー名は「JMarsh」です。 パスワード を省略すると、パスワードの入力を要求されます。この例では、パスワードは「password」です。ホスト名を省略すると、デフォルトで localhost となります。この例では、ホスト名は「orion」です。 ポート番号は必須です。この例では、デフォルトのポート番号 49201 を使用しています。

例

次の例は、stcmd add-project を使用して Orion というコンピュータ上に Integrations という名のプロジェクトを作成します（Orion は、ポート 1024 を使用するサーバー構成で StarTeam Server のインスタンスを実行しています）。このコマンドはプロジェクトを作成し、ワークステーションとサーバー間で送られるデータを圧縮して暗号化することを指定し、プロジェクトに説明を付けます。

```
stcmd add-project -s "JMarsh:password@Orion:1024" -cmp -encrypt "RC4" -name "Integrations"
-rp "C:¥integrations" -d "integrations between our products and our partner' s products"
```

関連参照

クライアントのコマンドライン操作

ビューの追加: stcmd add-view

stcmd add-view は、コマンドラインからサーバー構成にビューを追加するために使用します。ビューが作成されるとき、その親ビューは -p オプションで指定したビューで、そのルート フォルダは -p オプションで指定したフォルダです。このコマンドでは、-rp オプションはルート フォルダの作業フォルダを指定します。以下のオプションを使用して以下の種類のビューを作成します。

- -dr を使用して読み取り/書き込み参照ビューを作成します。
- -dr -ro を使用して読み取り専用参照ビューを作成します。
- -dr -ba を使用して、既存アイテムの動作が変更時分岐に設定されている分岐ビューを作成します。
- -dr -bn を使用して、既存アイテムの動作が変更時分岐に設定されていない分岐ビューを作成します。
- -dr を使用しないと空のビューが作成されます。

構文

このコマンドの構文は次のとおりです。

```
stcmd add-view -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-encrypt encryptionType]
[-q] [-x] [-stop] -name "viewName" [-rp "folderPath"] [-d "description"] [-dr [-ro | -ba | -bn
[-cfgl "labelName" | -cfgp "stateName" | -cfgd "asOfDate"]]]
```

オプション	説明
-ba	-dr と一緒に使用して、既存アイテムの動作が変更時分岐に設定されている分岐ビューを作成します。ビューのプロパティ [ビューで共有されるアイテムを変更時分岐に設定する] はオンに設定されます。 このオプションは、-dr と一緒に使用しなければなりません。
-bn	-dr と一緒に使用して、既存アイテムの動作が変更時分岐に設定されていない分岐ビューを作成します。ビューのプロパティ [ビューで共有されるアイテムを変更時分岐に設定する] はオフに設定されます。このオプションは、-dr と一緒に使用しなければなりません。
-cfgd	指定した日付/時刻のものとしてビューを構成します。日時指定の例を以下に示します。 "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST" このオプションは、次のいずれかと一緒に使用しなければなりません: -dr -ro、-dr -ba、または、-dr -bn。
-cfgl	指定されたラベルを使用してビューを構成します。-cfgl、-cfgp、または、-cfgd を指定していなければ、ビューの現在の構成が使用されます。このオプションは、次のいずれかと一緒に使用しなければなりません: -dr -ro、-dr -ba、または、-dr -bn。
-cfgp	指定されたプロモーション状態を使用してビューを構成します。このオプションは、次のいずれかと一緒に使用しなければなりません: -dr -ro、-dr -ba、または、-dr -bn。
-d	ビューに対する説明を指定します。最大 254 文字まで入力できます。
-dr	派生ビューを指定します。ブランク ビューを除くすべてのビューは、派生させることができます。-ba、-bn、および、 -ro も参照してください。 -ba、-bn、-ro を付けずに使用すると、読み取り/書き込み可能な参照ビューが作成されます。読み取り/書き込み可能な参照ビューの構成は、親ビューの構成と同じです。このため、-dr オプションを -ba、-bn、-ro を付けずに使用して、-cfgl、-cfgp、または -cfgd を付けると、エラー メッセージが出力されます。 このオプションを使用しないと、ブランク ビューが作成されます。ブランク ビューの場合は、ビューのプロパティ [ビューで共有されるアイテムを変更時分岐に設定する] はオフに設定されます。

-name	ビューの名前を指定します。最大 254 文字まで入力できます。
-ro	-dr と一緒に使用する場合は、読み取り専用の参照ビューを指定します。

例

次の例は、stcmd add-view を使用して Orion という名のコンピュータ上に Maintenance 5.1 という名の分岐ビューを作成します（Orion は、ポート 1024 を使用するサーバー構成で StarTeam Server のインスタンスを実行しています）。

このコマンドは、ビューを既存の StarDraw ビューの子として作成し、StarDraw フォルダをそのルート フォルダとして使用します。新規ビューは、5.1 製品が出荷される前の最後のビルド（ビルド 403）に使用されているラベルを基にしています。これは親の作業フォルダとは異なる作業フォルダを持ちます。ビュー内のすべての既存アイテムの動作は、変更時分岐に設定されます。

```
stcmd add-view -p "JMarsh:password@Orion:1024/StarDraw/StarDraw/" -cmp -encrypt "RC4" -name
"Maintenance 5.1" -rp "C:¥StarDraw¥Maintenance 5.1" -d "Maintenance view for 5.1 release of our product"
-dr -ba -cfgl "Build 403"
```

以下の例は、stcmd add-view を使用して、Orion という名前のコンピュータ上に Rooted At Source Code という名前の読み取り/書き込み参照ビューを作成します。このコマンドは、ビューを既存の StarDraw ビューの子として作成し、SourceCode フォルダをそのルート フォルダとして使用します。これは親と同じ作業フォルダを持ちます。読み取り/書き込み参照ビューは、親と同じ構成のはずなので、-cfgl、-cfgp、および -cfgd オプションは使用できません。

```
stcmd add-view -p "JMarsh:password@Orion:1024/ StarDraw/StarDraw/Source Code"
-cmp -encrypt "RC4" -name "Rooted At SourceCode" -d
"StarDraw main view but with SourceCode folder as the root of the hierarchy" -dr
```

関連参照

クライアントのコマンドライン操作

ラベルの適用: stcmd apply-label

stcmd apply-label は、指定したファイル リビジョンにビュー ラベルまたはリビジョン ラベルを付けるために使用します。ラベルは既に StarTeam に存在していなければなりません。ラベルは StarTeam 内または stcmd label コマンドで作成できます。

構文

このコマンドの構文は次のとおりです。

```
stcmd apply-label -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-csf]
[-encrypt encryptionType] [-is] [-q] [-x] [-stop] [-rp "folderPath" | -fp "folderPath"]
[-filter "fileStatus"] [-vl "labelName" | -vd "asOfDate" | -vn revisionNumber] -lbl "labelName" [files...]
```

オプション	説明
-filter	1 つまたは複数の文字から成る文字列を指定します。各文字が、ファイル ステータスを表します。この文字列には、スペースやその他のホワイトスペースは入れないで下さい。ラベルは、現在指定されているステータスを持つファイルにのみ適用されます。「ビュー外」のファイルにラベルを適用することはできません。ステータスを表すために使用できる文字は、次のとおりです: C(最新)、M(変更済み)、G(マージ)、O(古いリビジョン)、I(作業ファイルなし)、U(不明)。
-lbl	指定されたリビジョンに付加するラベル名を指定します。このオプションは複数回使用できます。アプリケーションは、すべてのラベルを、指定されたファイルまたはリビジョンに付加します。
-vd	新規ラベルを付加するリビジョンを識別するために使用する日時を指定します。日時指定の例を以下に示します。 "12/29/01 12:41 PM" "December 29, 2001 12:41:21 PM" "Monday, December 29, 2001 12:41"
-vl	新規ラベルを付加するリビジョンを識別するために使用するリビジョン ラベルまたはビュー ラベルを指定します。ラベルは既にアプリケーションに存在していなければなりません。-vn、-vd、または -vl オプションを指定しない場合、チップ リビジョンのファイルがチェックアウトされます。
-vn	新規ラベルを付加するリビジョンを識別するために使用するリビジョン番号を指定します。

例

以下の例は、stcmd apply-label を使用してルート フォルダ StarDraw(StarDraw プロジェクトの StarDraw ビュー内)の子である User Manual にラベル Beta を付けます。StarTeam は、2003 年 7 月 7 日の正午に最新であったファイルのリビジョンにラベルを付けます。

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

関連参照

クライアントのコマンドライン操作

ファイルのチェックイン：stcmd ci

stcmd ci は、コマンドラインを使用して作業フォルダから StarTeam リポジトリ（データ保管庫）にファイルをチェックインするために使用します。

同時に、新規ファイル リビジョンを処理アイテムにリンクすることができます。このコマンドを使用して正常に追加されたファイルは、すべて処理アイテムのチップ リビジョンにリンクされ固定されます。-active オプションを使用して現在アクティブな処理アイテム（以前に、ユーザーのワークステーションで StarTeam クライアントを使用して設定された）を指定します。

アクティブなアイテムがないか、または別なアイテムを使用する場合、処理アイテムのタイプを示すオプション（-cr、-req、-task）に続いてプロジェクト ビューのルート フォルダからアイテムまでの完全なパスを使用します。この際、フォルダ名間の区切りにはスラッシュ（/）を使用します。ビュー外の処理アイテムについては、完全なフォルダ パスの前にプロジェクト名とビュー名を指定します。ビュー パスは、コロン（:）で区切ります。たとえば、 -cr MyProject/RootView:ChildView/SourceCode/37 は、MyProject プロジェクトの ChildView ビューの SourceCode フォルダにある変更要求 37 を指定します。実行中、処理ではまず処理アイテムが現在のビューにあるものとみなし、現在のビューを確認してフル パスがそのビュー内のフォルダ パスに対応するかどうか判断します。処理アイテムが現在のビューにない場合、処理外アイテムとして扱われ、プロジェクトとビューから処理アイテムの検索を開始します。

同時に、-mark オプションを使用して処理アイテムをそのタイプに従って「解決」、「終了」、「完了」とマークすることもできます。たとえば、変更要求は解決とマークできます。アイテムは、すべてのファイルが正常に追加されなかった場合は、「解決」、「終了」、「完了」とマークされません。

構文

このコマンドの構文は次のとおりです。

```
stcmd ci -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-csf] [-encrypt encryptionType]
[-is] [-q] [-x] [-stop] [-rp "folderPath" | -fp "folderPath"] [-filter "fileStatus"] [-l | -u | -nel]
[-ro | -rw]] [-vl "labelName"] [-nomove] [-f NCI] [-o | -i ] [-r "comment" | -rf " fileName "] [-eol
[on | off]] [[ -active | [-cr | -req | -task ] processItemPath] [-mark]] [files...]
```

オプション	説明
-active	アクティブな処理アイテムを指定します。
-cr -req -task	プロジェクト ビューのルート フォルダから処理アイテムとして使用する変更要求、要件、またはタスク番号への完全なパスを指定します。フォルダ名の区切り文字にはスラッシュ（/）を使用します。 ビュー外の処理アイテムについては、完全なフォルダ パスの前にプロジェクト名とビュー名を指定します。ビュー パスは、コロン（:）で区切ります。たとえば、 -cr MyProject/RootView:ChildView/SourceCode/37 は、MyProject プロジェクトの ChildView ビューの SourceCode フォルダにある変更要求 37 を指定します。 実行中、処理ではまず処理アイテムが現在のビューにあるものとみなし、現在のビューを確認してフル パスがそのビュー内のフォルダ パスに対応するかどうか判断します。処理アイテムが現在のビューにない場合、処理外アイテムとして扱われ、プロジェクトとビューから処理アイテムの検索を開始します。
-eol	作業中のテキスト ファイルをサーバーのリポジトリに転送する際に、行末マーカー（EOL）を復帰/改行（CR/LF）に自動変換します。このオプションが on の場合、作業ファイルの行末マーカー（EOL）を復帰/改行（CR/LF）に変換します。デフォルトでは、このオプションは off になっており、行末変換は行われません。Windows クライアントでは、行末マーカー（EOL）は CR/LF、UNIX プラットフォームでは LF です。 例えば、UNIX で稼働しているコンピュータから作業ファイルをリポジトリにチェックインする際に、テキストファイルを CR/LF として保存したい場合は、このオプションを on にします。
-f NCI	「変更済み」ステータスを持つファイルをチェックインします。NCI は「チェックインが必要（Needs Check-In）」の略です。その他の種類のファイルはチェックインされません。 -filter と同時に指定した場合、-f NCI は無視されます。
-filter	1 つまたは複数の文字から成る文字列を指定します。各文字が、ファイル ステータスを表します。この文字列には、スペースやその他のホワイト スペースは入れないでください。現在、指定されたステータスを持っているファイルだけがチェックインされます。「ビュー外」のファイルをチェックインすることはできません。

	ステータスを表すために使用できる文字は、次のとおりです。 C(最新)、M(変更済み)、G(マージ)、O(古いリビジョン)、I(作業ファイルなし)、U(不明)。 -filter は、-f NCI よりも優先されます。G、O、または、U を使用する場合は、-i または -o も指定する必要があります。そうでなければ、G、O または U の指定は無視されます。
-i	通常はチェックインできないステータスを持つファイルを、対話的にチェックインできるようにします。「マージ」、「古いリビジョン」、「不明」ステータスを持つファイルが見つかるたびに、ダイアログが表示されます。ここで了承すれば、強制的にファイルをチェックインできます。 -i オプションを使用する場合は、-o オプションを使用できません。
-l	チェックインしたファイルをすべてロックします。-l、-u、または -nel を指定しない場合、ファイルのロック ステータスは変わりません。
-mark	すべてのファイルが正常に追加された場合に、処理アイテムのステータスを「解決」(変更要求の場合)、「終了」(タスクの場合)、「完了」(要件の場合)に変更します。ファイルは、新しいステータスを持つリビジョンに固定されます。
-nel	チェックインしたファイルすべてに非排他的ロックをします。
-nomove	チェックイン中のファイルが既に同じラベルのリビジョンを持つ場合に、-vl オプションで指定されたラベルの適用を中止します。それ以外の場合は、現在のラベルが付いているリビジョンから新規にチェックインしたリビジョンにラベルが移動します。
-o	通常はチェックインできないステータスを持つファイルを、強制チェックインします。このオプションは、「マージ」、「古いリビジョン」、「不明」ステータスのすべてのファイルを、強制的にチェックインします。 -o オプションを使用する場合は、-i オプションは使用できません。
-r	リビジョン コメントを指定します。通常これは、ファイルをチェックインする時の理由を表します。-r オプションを使用する場合は、-rf オプションを使用できません。
-rf	リビジョン コメントを含むファイルへのパスを指定します。 -rf オプションを使用する場合は、-f オプションを使用できません。
-ro	この操作の後、作業ファイルを読み取り専用にします。このオプションを指定しないと、ファイルは操作の前と同じ状態のままとなります。 通常は、-ro を使用して、自分がロックしていないファイルを自分自身が編集することを防ぎます。 -ro は、-l、-u、または -nel を一緒に使用しなければなりません。-ro を使用する場合は、-rw は使用できません。
-rw	この操作の後、作業ファイルを読み取り/書き込み可能にします。このオプションを指定しないと、ファイルは操作の前と同じ状態のままとなります。 -rw は、-l、-u、または -nel を一緒に使用しなければなりません。-rw を使用する場合は、-ro は使用できません。
-u	新規にチェックインしたファイルのロックを解除します。
-vl	チェックインしたファイルに適用するラベル(stcmd label を使用して作成された)を指定します。ラベルは二重引用符で囲みます。このオプションは、コマンド内で複数回指定できます。ここで指定できるラベルはビュー ラベルまたはリビジョン ラベルです。ただし、アプリケーション内に既に存在するラベルでなければなりません。

例

以下の例は、stcmd ci を使用して、ルート フォルダ StarDraw(StarDraw プロジェクトの StarDraw ビュー内)の子である Online Help に .bmp ファイルをチェックインします。このコマンドはファイルをロック解除し、作業コピーを読み取り専用にし、ファイル リビジョンにコメント(通常はファイルをチェックインする理由)を付けます。

```
stcmd ci -p "JMarsh:password@Orion:1024/StarDraw/StarDraw/SourceCode/Online Help" -u -ro -r
"revised for beta" "*.bmp"
```

関連参照

- クライアントのコマンドライン操作

ファイルのチェックアウト: stcmd co

stcmd co は、StarTeam のリポジトリ(データ保管庫)のファイルを自分の作業フォルダにチェックアウトするためにコマンドラインで使用します。このため、-o を使用しないと、「変更」、「マージ」、または「不明」ステータスを持つファイルが見つかるたびに、そのファイルがチェックアウトされないことを知らせるために一時停止します。

-merge オプションを付けると、チェックアウト処理の一部としてファイルをマージすることができます。マージはチェックイン処理の一部ではありません。

構文

このコマンドの構文は次のとおりです。

```
stcmd co -p "project" [-pwdfile "filename"] [-epwdfile "filename"] [-cmp] [-encrypt RC4, RC2_ECB, RC2_CBC,
RC2_CFB]
[-cfgl "label" | -cfgp "promotion state" | -cfgd "date"] [-is] [-csf] [-q] [-x] [-stop]
[-rp "directory" | -fp "directory"] [-filter "filter"][-?] [-h] [nologo] [-o | -i | -merge] [-hook
"executable"]
[-l | -u | -nel] [-ro | -rw]] [-vl "name" | -vd "date" | -vn number]
[-f NCO]  [-ts] [-eol [on | off | cr| lf| crlf ]] [-fs] [-dryrun | -alwaysprompt | -neverprompt | -
conflictprompt] [-mpxCacheAgent number]
[-useMPXCacheAgent  “host” port”  | autolocate]]  [files...]
```

オプション	説明
-cfgd	指定した日付/時刻のものとしてビューを構成します。日時指定の例を以下に示します。 "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
-cfgl	指定されたラベルを使用してビューを構成します。-cfgl、-cfgp、または、-cfgd を指定していなければ、ビューの現在の構成が使用されます。
-cfgp	指定されたプロモーション状態を使用してビューを構成します。
-eol	改行コードを自動的に変換します。 このオプションを指定すると、テキストファイルがサーバーのリポジトリからワークステーションの作業フォルダへと転送されるときに、コマンドを実行しているプラットフォームに合わせて Java VM により指定されたとおりに行末(EOL)が変換されます。 デフォルトでは、このオプションは off になっており、行末変換は行われません。off と指定するのと -eol を指定しないのとでは、結果は同じです。 改行文字(cr、lf、crlf のいずれか)を指定すると、テキストファイルがサーバーのリポジトリからワークステーションの作業フォルダに転送されるときに、指定した EOL 変換が行われます。 行末マーカー(EOL)は、Windows Client では復帰/改行(CR/LF)、UNIX プラットフォームでは改行(LF)、Macintosh オペレーティングシステムでは復帰(CR)です。 例えば、リポジトリから取得したファイルを UNIX システム上の作業ファイルと比較する場合(リポジトリにはテキストファイルが crlf ファイルとして保存されている場合)、このオプションを on または lf に設定します。
-f NCO	「作業ファイルなし」または「古いリビジョン」ステータスを持つファイルをチェックアウトします。NCO は「チェックアウトが必要」という意味です。その他の種類のファイルはチェックアウトされません。 -filter と同時に指定した場合、-f NCO は無視されます。
-filter	1 つまたは複数の文字から成る文字列を指定します。各文字が、ファイル ステータスを表します。この文字列には、スペースやその他のホワイトスペースは入れないで下さい。現在、指定されたステータスを持っているファイルだけがチェックアウトされます。「ビュー外」のファイルをチェックアウトすることはできません。ステータスを表すために使用できる文字は、次のとおりです。C(最新)、M(変更済み)、G(マージ)、O(古いリビジョン)、I(作業ファイルなし)、U(不明)。

	-filter は、-f NCO よりも優先されます。G、M、O または U を指定する場合は、-merge または -o を使って、強制的にチェックアウトしなければなりません。これを行わないと、G、M、O または U の指定は無視されます。
-fs	チェックアウト後にファイル ステータスが記憶されないようにします。これらのファイルの以降のステータス値は、正しくなく不定になります。ファイルのステータスが重要ではない場合に、このオプションを使用します。たとえば、いつも作業フォルダを削除してからビルドのためにファイルをチェックアウトしている場合、ファイルが存在しないので、そのステータスは問題になりません。 このファイル ステータスは、たとえ後から update-status コマンドを使ったとしても既知にはなりません。-fs オプションを指定せずにチェックアウトを強制すれば、正しいステータスを持つ最新のファイルが得られます。
-hook	-merge と一緒に使用します。マージを実行するために、別のアプリケーション(アプリケーションでは利用できないアプリケーション)を指定します。 このオプションの値は、マージを実行するために起動するプログラム名にします。例えば、次のように指定します: -hook mymerge.sh マージ アプリケーションは、競合を検出しなかったことを意味する終了コード 0、または競合を検出したことを意味する終了コード 1 を返す必要があります。それ以外の値はエラーを表します。Windows プラットフォームでは、バッチ ファイルをマージ アプリケーションにすることはできません。Java を介して実行すると、終了コードが正しく戻らないからです。 マージするファイルごとに、stcmd はマージ アプリケーションに 3 つの引数を渡します。この引数は、以下の 3 つのファイル(順番も以下のとおり)への完全なパスです。 localFile: ローカルの作業ファイル commonFile: ローカルに変更されたファイルと、リポジトリ内の指定されたリビジョンの間の共通な最新リビジョンの内容 otherFile: チェックアウトされているファイルリビジョンの内容 マージ用のフック プログラムは、マージ結果を標準出力(stdout)に出力する必要があります。
-i	通常はチェックアウトできないステータスを持つファイルを、対話的にチェックアウトできるようにします。「変更済み」、「マージ」、「不明」ステータスを持つファイルが見つかるたびに、ダイアログが表示されます。ここで了承すれば、強制的にファイルをチェックアウトできます。-i オプションを使用する場合は、-o または -merge オプションは使用できません。
-l	チェックアウトしたファイルをすべてロックします。-l、-u、または -nel を使用しない場合は、ファイルのロック状態は以前のままです。
-merge	作業ファイルとチェックアウトされたリビジョンをマージしたり、作業ファイルとその基になったリビジョンをマージします。この作業ファイルは、「マージ」ステータスを持ちます。 -merge と一緒に、以下のレポート オプションのいずれかを指定できます。 -dryrun: マージ結果ファイルが競合を持つかどうかを示します。何もチェックアウトされないので、ローカルな作業ファイルは変更されません。これはプレビューを提供します。 -alwaysprompt: マージの際の競合の有無に関わらず、マージ結果ファイルを作業フォルダに保存するかどうかの確認をユーザーに要求します。 -neverprompt: マージ結果ファイルを常に作業フォルダに保存します。 -conflictprompt: 競合が検出された場合のみ、マージ結果ファイルを保存するかどうかの確認をユーザーに要求します。 -dryrun、-alwaysprompt、-neverprompt、-conflictprompt の各オプションは、互いに排他的です。レポートオプションを何も指定しないと、デフォルトで -conflictprompt と同じになります。 マージを実行するために、別のアプリケーション(アプリケーションでは利用できないアプリケーション)を指定することもできます。 -merge オプションを使用する場合は、-i または -o オプションは使用できません。 -hook を使用せずに、競合を持つマージ結果ファイルを保存すると、作業ファイル内では、競合部分に次のようなマークが付きます。 <<<<<<< fileName (local) ローカル ファイルにあった行 =======

	チェックアウトされたリビジョンにあった行 >>>>>>> fileName (version x) ここでの fileName はファイル名、x はチェックアウトされたリビジョン番号です。 -merge を指定しない場合は、-o を指定して強制的にチェックアウトしない限り、「マージ」ステータスを持つファイルは無視されます。
-nel	チェックアウトしたファイルすべてに非排他的ロックをします。
-o	どんなステータスのファイルも強制的にチェックアウトします。-o オプションを使用する場合は、-i または -merge オプションは使用できません。
-ro	この操作の後、作業ファイルを読み取り専用にします。このオプションを指定しないと、ファイルは操作の前と同じ状態のままとなります。通常は、 -ro を使用して、自分がロックしていないファイルを自分自身が編集することを防ぎます。 -ro は、-l、-u、または -nel を一緒に使用しなければなりません。-ro を使用する場合は、-rw は使用できません。
-rw	この操作の後、作業ファイルを読み取り/書き込み可能にします。このオプションを指定しないと、ファイルは操作の前と同じ状態のままとなります。 -rw は、-l、-u、または -nel を一緒に使用しなければなりません。-rw を使用する場合は、-ro は使用できません。
-ts	各作業フォルダのタイムスタンプをチェックアウトの時刻に設定します。このオプションを指定しないと、ファイルには、そのファイルのチェックイン リビジョンと同じタイム スタンプが与えられます。
-u	チェックアウトしたファイルをロック解除します。
-vd	チェックアウトするリビジョンを識別するために使用する日時を指定します。指定した日時以前の最新リビジョンが、すべてのファイルに対してチェックアウトされます。日時指定の例については、上記の -cfgd を参照して下さい。
-vl	チェックアウトするリビジョンを識別するために使用するリビジョンラベルまたはビューラベルを指定します。-vn、-vd、または、-vl オプションを指定しないと、チップリビジョンのファイルがチェックアウトされます。
-vn	チェックアウトするファイルのリビジョン番号を指定します。

例

以下の例は、stcmd co を使用して、ルート フォルダ StarDraw(StarDraw プロジェクトの StarDraw ビュー内)の子である User Manual から .doc ファイルをロックしてチェックアウトします。

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

次の例は、stcmd co を使用して readme ファイルをマージします。

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

関連参照

クライアントのコマンドライン操作

ファイル リビジョン間の比較: stcmd diff

stcmd diff を使用してファイルの 2 つのリビジョン間の差分を表示します。このコマンドは複数のファイルに適用できます。リビジョンを指定しないと(-vn または -vd または -vl を使用して)、指定した各ファイルの作業コピーが、このファイルのリポジトリ(データ保管庫)内のそのファイルのチップ リビジョンと比較されます。1 つのリビジョンを指定すると、指定した各ファイルの作業コピーがそのリビジョンと比較されます。2 つのリビジョンを指定すると、指定した各ファイルのその 2 つのリビジョンが比較されます。

テキストファイルを比較する場合は、差分を表示することができます。バイナリ ファイルを比較すると、そのリビジョンのファイルが同じか違うかを表す結果が出力されます。

構文

このコマンドの構文は次のとおりです。

```
stcmd diff -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-csf]
[-encrypt encryptionType] [-cfgl "labelName"| -cfgp "stateName" | -cfgd "asOfDate"]
[-is] [-q] [-x] [-stop] [-rp "folderPath" | -fp "folderPath"] [-filter "fileStatus"]
[-eol [on | off | cr| lf| crlf]] [-w | -Bpvcs | -b] [-i] [-m "maskSet"] [-t number] [-c number] [-n]
[-nd] [-e] [-vl "labelName" | -vd "asOfDate" | -vn revisionNumber] [files...]
```

オプション	説明
-b	テキストファイルの 2 つの行を比較するときに、行の最後に付いているホワイトスペースを無視し、それ以外のホワイトスペースの文字列はすべて同じ長さであるものとして扱います。例えば、次の行は同じと見なされます。 " hi mom " " hi mom"
-Bpvcs	テキスト ファイルの 2 つの行を比較するときに、最初と最後のホワイトスペースを無視します。例えば、以下の行は "hi" と "mom"の間に空白が 1 つしかないので、同じと見なされます。 " hi mom " " hi mom" ただし、次の行は同じとは見なされません。 "hi mom"
-c	テキストファイル内に差分が見つかった前と後に表示する、変更されていない行の行数を指定します。このオプションを指定しないと、ファイルのすべての行が表示されます。例えば、-c 2 と指定すると、変更された行の前と後に、変更されていない行が 2 行配置されます。
-cfgd	指定した日付/時刻のものとしてビューを構成します。日時指定の例を以下に示します。 "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
-cfgl	指定されたラベルを使用してビューを構成します。-cfgl、-cfgp、または、-cfgd を指定していなければ、ビューの現在の構成が使用されます。
-cfgp	指定されたプロモーション状態を使用してビューを構成します。
-e	このコマンドによって、次の終了コードが返るようになります。 0 は、比較したファイルがすべて等しい場合 1 は、エラー条件が発生した場合 2 は、少なくとも 1 つのファイルが異なる場合

	-e は、テキストファイルまたはバイナリ ファイルに使用します。
-eol	このコマンドは、diff が行末マーカー(EOL)を無視している場合には意味がありません。行末マーカーを無視している場合を除いて、2 つの行が同じであれば、同じであると報告します。
-filter	1 つまたは複数の文字から成る文字列を指定します。各文字が、ファイル ステータスを表します。この文字列には、スペースやその他のホワイト スペースは入れないでください。現在、指定されたステータスを持っているファイルだけが比較されます。「ビュー外」のファイルのリビジョンを比較することはできません。 ステータスを表すために使用できる文字は、次のとおりです。C(最新)、M(変更済み)、G(マージ)、O(古いリビジョン)、I(作業ファイルなし)、U(不明)。
-i	2 つのテキストファイルを比較するときに、大文字小文字を無視します。例えば、"A" は "a"と同じと見なされます。
-m	2 つのテキストファイルを比較するときに、1 つ以上のマスクで指定された特定の列の文字を無視します。マスクには次の構文を使用します。 "columnNumber-columnNumber[(numeric)]" 例えば、"1-6" は、各行の最初の 6 列に含まれる文字を無視します。"1-6(numeric)" は、両方のファイルの 1 列目の文字が数字の場合は、両方の行の最初の 6 列を無視します。 複数のマスクを使用する場合は、それぞれのマスクをカンマで区切ります。構文は次のとおりです。 "mask[,mask]..."
-n	2 つのテキスト ファイルの行番号を表示しないようにします。
-nd	2 つのテキストファイルの差分を表示しないようにします。バイナリファイルを比較する場合は、差分は表示されません。このオプションは -e オプションと組み合わせて使用すると便利です。
-t	テキスト ファイルの差分を表示するときに使用するタブ ストップの空白数を指定します。デフォルトは 4 です。-t 0 を指定すると、タブ変換は行われません。
-vd	比較するリビジョンを識別するために使用する日時を指定します。指定した日時以前の最新リビジョンが使われます。日時指定の例については、上記の -cfgd を参照して下さい。
-vl	比較するリビジョンを識別するために使用するリビジョンラベルまたはビューラベルを指定します。-vn、-vd、または -vl のいずれか 1 つ、または 2 つを組み合わせて使用できます。れらのオプションを全く使用しなくてもかまいません。 オプションを何も指定しない場合は、作業ファイルがチップリビジョンと比較されます。1 つ指定すると、作業ファイルが指定したリビジョンと比較されます。2 つ指定すると、指定した 2 つのリビジョンが比較されます。
-vn	比較するリビジョン番号を指定します。
-w	テキストファイルの 2 つの行を比較するときに、ホワイトスペース(タブとスペース)を無視します。例えば、次の行は同じと見なされます。 " a = (b + 2);" "a=(b+2);" -w、-Bpvcs、-b の各オプションは、互いに排他的です。

例

以下の例は、stcmd diff を使用して、フォルダ SourceCode 内にある .cpp ファイルそれぞれに対して、リビジョン Beta1 と Beta2 を比較しています。このフォルダは、ルート フォルダ StarDraw(StarDraw プロジェクトの StarDraw ビュー内)の子フォルダです。空白は無視します。

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

関連参照

- クライアントのコマンドライン操作

ファイル説明の変更：stcmd dsc

stcmd dsc は、コマンドラインからファイルの説明を変更するために使用します。このコマンドは、プロパティの 1 つに新しい説明を入れて新しいファイル リビジョンを作成します。

構文

このコマンドの構文は次のとおりです。

```
stcmd dsc -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-csf] [-encrypt encryptionType]
[-is] [-q] [-x] [-stop] [-rp "folderPath" | -fp "folderPath"] [-filter "fileStatus"] -d
"description" [files...]
```

オプション	説明
-d	ファイルに対する説明を指定します。この文章は二重引用符で囲みます。
-filter	1 つまたは複数の文字から成る文字列を指定します。各文字が、ファイル ステータスを表します。この文字列には、スペースやその他のホワイトスペースは入れないで下さい。現在、指定されたステータスを持っているファイルだけに説明が付加されます。「ビュー外」のファイルの説明を変更することはできません。ステータスを表すために使用できる文字は、次のとおりです。C（最新）、M（変更済み）、G（マージ）、O（古いリビジョン）、I（作業ファイルなし）、U（不明）。

例

以下の例は、stcmd dsc を使用して、ルート フォルダ StarDraw（StarDraw プロジェクトの StarDraw ビュー内）の子である User Manual 内の stdafx.cpp の説明を変更します。

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

関連参照

クライアントのコマンドライン操作

ラベルの作成: stcmd label

stcmd label を使用してビュー ラベルまたはリビジョン ラベルを作成します。ビュー ラベルをビルド ラベルとして指定することができます。デフォルトで、ビュー ラベルはビュー内の各フォルダ、ファイル、変更要求、要件、トピック、およびタスクに自動的に適用されます。デフォルトで、リビジョン ラベルはアイテムには適用されません。

stcmd apply-label を使用して stcmd label で作成したラベルを指定ファイルに適用することができます。また、stcmd ci でラベル オプション(-vl)を使用してファイルをチェックインするとき新規ラベルを付けることもできます。

オプション	説明
-b	新規ラベルをビルド ラベルに指定します。-b または -r を使用しないと、を使用しないと、ラベルはビューラベルになります。ビュー ラベル(ビュー ラベルの特殊形であるビルドラベルも含む)は、直ちにビュー内のすべてのフォルダ、ファイル、変更要求、タスク、およびトピックに自動的に適用されます。
-d	ラベルの説明を指定します。
-f	新規ラベルを凍結されたラベルとして作成します。
-nl	新規ラベルの名前を指定します。
-r	新規ラベルをリビジョン ラベルに指定します。新規ラベルを使って、チェックインするファイルにラベルを付けることができます。このコマンドでは、いくつかのアイテムに付加されている既存のリビジョンラベルをコピーしてラベルを作成する場合以外は、新規ラベルをアイテムに付加することができません。-vl オプションについては、以下を参照してください。
-vd	ビュー ラベルを作成するときに、そのラベルに対する日付/時刻を指定します。リビジョン ラベルを作成する場合、-vd オプションは無視されます。-vn、-vd、または、-vl オプションを指定しないと、ビュー ラベルには現在の日時が使われます。日時指定の例を以下に示します。 "12/29/01 12:41 PM" "December 29, 2001 12:41:21 PM" "Monday, December 29, 2001 12:41"
-vl	コピーするラベルの名前を指定します。ラベルの名前は二重引用符で囲みます。指定したラベルは既にアプリケーション内に存在しなければなりません。また、これから作成するラベルと同じ種類でなければなりません。指定したラベルが、いくつかのアイテムに付加されているリビジョンラベルの場合は、新規ラベルはこれらと同じアイテムに付加されます。 -vn、-vd、または、-vl オプションを指定しないと、ビュー ラベルには現在の日時が使われます。 リビジョンラベルの場合は、必ず -r を使用して下さい。
-vp	コピーするラベルを持つプロモーション状態を指定します。状態の名前は二重引用符で囲みます。指定したラベルは、既にアプリケーション内に存在していなければなりません。このオプションで作成するビューラベルは有効でなければなりません。 -vn、-vd、または、-vl オプションを指定しないと、ビュー ラベルには現在の日時が使われます。 プロモーション状態があるラベルに設定されており、最新 には設定されていない場合、プロモーション状態を元にしてのみ、ビュー ラベルを作成することができます。

構文

このコマンドの構文は次のとおりです。

```
stcmd label -p "username:password@host:port/project/view/folder" [-pwdfile "filePath"] [-cmp] [-encrypt
encryptionType]
[-q] [-x] [-stop] -nl "labelName" [-vl "labelName" | -vd "asOfDate" | -vp stateName]
[-d "description"] [-b | -r] [-f]
```

例

以下の例は、stcmd label を使用して StarDraw プロジェクトの StarDraw ビューに Beta という名前の新規ビルド ラベルを作成します。

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

関連参照

クライアントのコマンドライン操作

作業フォルダの作成: stcmd local-mkdir

stcmd local-mkdir を使用してユーザーのワークステーション上に指定した StarTeam フォルダの作業フォルダまたは作業ディレクトリを作成します。指定した StarTeam フォルダの子フォルダの作業フォルダ(作業ディレクトリ)も作成するには -is を使用します。

構文

このコマンドの構文は次のとおりです。

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

オプション	説明
-cfgd	指定した日付/時刻のものとしてビューを構成します。日時指定の例を以下に示します。 "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
-cfgl	指定されたラベルを使用してビューを構成します。-cfgl、-cfgp、または、-cfgd を指定していなければ、ビューの現在の構成が使用されます。
-cfgp	指定されたプロモーション状態を使用してビューを構成します。

例

以下の例は、stcmd local-mkdir を使用してルート フォルダ StarDraw(StarDraw プロジェクトの StarDraw ビュー内)の子である SourceCode とその子フォルダの作業フォルダを作成します。

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

関連参照

クライアントのコマンドライン操作

ローカル ファイルの削除: stcmd delete-local

stcmd delete-local は、作業フォルダから作業ファイルを削除するために使用します。バージョン管理下に置かれているファイルおよび StarTeam 内にないファイルを削除することができます。この動作は、バージョン管理からファイルを取り除きません。単にユーザーのワークステーションで作業フォルダに保管されているデータの量を削減するだけです。アプリケーションのステータスに基づいてファイルを削除する場合、まず stcmd update-status を使用することを推奨します。

構文

このコマンドの構文は次のとおりです。

```
stcmd delete-local -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-csf]
[-encrypt encryptionType] [-is] [-q] [-x] [-stop] [-rp "folderPath" | -fp "folderPath"]
[-cfgl "labelName" | -cfgp "stateName" | -cfgd "asOfDate"] [-filter "fileStatus"] [files...]
```

オプション	説明
-cfgd	指定した日付/時刻のものとしてビューを構成します。日時指定の例を以下に示します。 "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
-cfgl	指定されたラベルを使用してビューを構成します。-cfgl、-cfgp、または、-cfgd を指定していなければ、ビューの現在の構成が使用されます。
-cfgp	指定されたプロモーション状態を使用してビューを構成します。
-filter	1 つまたは複数の文字から成る文字列を指定します。各文字が、ファイル ステータスを表します。この文字列には、スペースやその他のホワイトスペースは入れないで下さい。現在、指定されたステータスを持っているファイルだけが削除されます。ステータスを表すために使用できる文字は、次のとおりです: C(最新)、M(変更済み)、G(マージ)、O(古いリビジョン)、I(作業ファイルなし)、U(不明)。

例

以下の例は、stcmd delete-local を使用して SourceCode という名の StarTeam フォルダの作業フォルダからファイルをいくつか削除します。SourceCode は、ルート フォルダ StarDraw(StarDraw プロジェクトの StarDraw ビュー内)の子です。この例はバージョン管理下に置かれていないすべてのファイルを削除します。これらのファイルは、**Not In View** というファイル ステータスです。

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

関連参照

クライアントのコマンドライン操作

ファイル履歴の表示: stcmd hist

stcmd hist を使用してファイルのリビジョン履歴を表示します。

構文

このコマンドの構文は次のとおりです。

```
stcmd hist -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-csf]
[-encrypt encryptionType] [-cfgl "labelName" | -cfgp "stateName" | -cfgd "asOfDate"]
[-is] [-q] [-x] [-stop] [-rp "folderPath" | -fp "folderPath"] [-filter "fileStatus"] [files...]
```

オプション	説明
-cfgd	指定した日付/時刻のものとしてビューを構成します。日時指定の例を以下に示します。 "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
-cfgl	指定されたラベルを使用してビューを構成します。-cfgl、-cfgp、または、-cfgd を指定していなければ、ビューの現在の構成が使用されます。
-cfgp	指定されたプロモーション状態を使用してビューを構成します。
-filter	1 つまたは複数の文字から成る文字列を指定します。各文字が、ファイル ステータスを表します。この文字列には、スペースやその他のホワイトスペースは入れないで下さい。現在、指定されたステータスを持っているファイルだけが報告されます。「ビュー外」のファイルの履歴を表示することはできません。 ステータスを表すために使用できる文字は、次のとおりです。C(最新)、M(変更済み)、G(マージ)、O(古いリビジョン)、I(作業ファイルなし)、U(不明)。

例

以下の例は、stcmd hist を使用して、ルート フォルダ StarDraw(StarDraw プロジェクトの StarDraw ビュー内)の子である SourceCode 内のファイル star.h のリビジョン履歴を表示します。

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

関連参照

クライアントのコマンドライン操作

ファイルの一覧表示: stcmd list

stcmd list を使用して -p オプションで指定したフォルダ内のすべてのファイルのリストを表示します。ファイルは、特定の時点で存在したファイルまたは特定のラベルを持つファイルです。

構文

このコマンドの構文は次のとおりです。

```
stcmd list -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-csf]
[-encrypt encryptionType] [-cfgl "labelName" | -cfgp "stateName" | -cfgd "asOfDate"]
[-is] [-q] [-x] [-stop] [-rp "folderPath" | -fp "folderPath"] [-filter "fileStatus"]
[-cf] [files...]
```

オプション	説明
-cf	フォルダの子フォルダの名前をリストに追加します。
-cfgd	指定した日付/時刻のものとしてビューを構成します。日時指定の例を以下に示します。 "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
-cfgl	指定されたラベルを使用してビューを構成します。-cfgl、-cfgp、または、-cfgd を指定していなければ、ビューの現在の構成が使用されます。-cfgp は、指定されたプロモーション状態を使用してビューを構成します。
-filter	1 つまたは複数の文字から成る文字列を指定します。各文字が、ファイル ステータスを表します。この文字列には、スペースやその他のホワイト スペースは入れないでください。現在、指定されたステータスを持っているファイルだけが一覧表示されます。「ビュー外」のファイルは一覧表示されません。 ステータスを表すために使用できる文字は、次のとおりです: C(最新)、M(変更済み)、G(マージ)、O(古いリビジョン)、I(作業ファイルなし)、U(不明)。
-short	ローカルファイルとそのステータスの一覧を簡潔な表示で出力します。ファイルのステータスは、ステータスを表す略字と作業ファイルへの相対パスで構成されます。以下に例を示します。 M /starteam/Server.java N /starteam/LabelInfo.java このオプションがない場合は、一覧は以下のものから構成されます。 フォルダの名前と作業フォルダのパスを表す 1 行。1 つのフォルダ内のファイルごとに、ステータスの略称、アクセス権限、タイムスタンプ、およびファイル名を表す 1 行。 例: Folder: Source (working dir: E:¥Source) Unknown rw 4/6/02 7:42:18 PM PST 230 req.bmp

例

以下の例は、stcmd list を使用して、ルート フォルダ StarDraw(StarDraw プロジェクトの StarDraw ビュー内)の子である SourceCode 内のすべてのファイルおよび SourceCode の子フォルダ内のすべてのファイルの一覧を表示します。

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

-short でローカル ファイルとそのステータスの一覧を簡潔な表示で出力します。ファイルのステータスは、ステータスを表す略字と作業ファイルへの相対パスで構成されます。以下に例を示します:

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

-short オプションを指定しない場合、各フォルダに付き、フォルダ名とそれに続いて対応する作業フォルダ パスが 1 行に記載されたデータで構成されたリストが作成されます。それぞれのフォルダ内には、各ファイルに付き、ステータスの略称、アクセス権限、タイムスタンプ、およびファイル名が 1 行に記載されたデータで構成されたリストが作成されます。例:

```
Folder: Source  (working dir: E:¥Source) Unknown rw 4/6/02 7:42:18 PM PST        230 req.bmp
```

関連参照

クライアントのコマンドライン操作

ファイルのロックとロック解除: stcmd lck

stcmd lck を使用してコマンドラインからファイルのロックまたはロック解除を行います。

オプション	説明
-break	ロックを強制解除する権限を持っている場合に、他人がかけたロックを強制解除します。
-filter	1 つまたは複数の文字から成る文字列を指定します。各文字が、ファイル ステータスを表します。この文字列には、スペースやその他のホワイト スペースは入れないでください。現在、指定されたステータスを持っているファイルだけがロックまたはロック解除されます。「ビュー外」のファイルをロックまたはロック解除することはできません。ステータスを表すために使用できる文字は、次のとおりです。C(最新)、M(変更済み)、G(マージ)、O(古いリビジョン)、I(作業ファイルなし)、U(不明)。
-l	ファイルをロックします。-l、-nel、または -u を使用しない場合は、これがデフォルトになります。
-nel	ファイルに対して非排他的ロックをします。
-ro	この操作の後、作業ファイルを読み取り専用にします。このオプションを指定しないと、ファイルは操作の前と同じ状態のままとなります。通常は、-ro を使用して、自分でロックしていないファイルを自分が編集することを防ぎます。 -ro は、-l、-u、または -nel を一緒に使用しなければなりません。-ro を使用する場合は、-rw は使用できません。
-rw	この操作の後、作業ファイルを読み取り/書き込み可能にします。このオプションを指定しないと、ファイルは操作の前と同じ状態のままとなります。 -rw は、-l、-u、または -nel を一緒に使用しなければなりません。-rw を使用する場合は、-ro は使用できません。
-u	ファイルをロック解除します。

構文

このコマンドの構文は次のとおりです。

```
stcmd lck -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-csf]
[-encrypt encryptionType] [-is] [-q] [-x] [-stop] [-rp "folderPath" | -fp "folderPath"]
[-filter "fileStatus"] [-break] [-l |-u | -nel] [-ro | -rw]] [files...]
```

例

以下の例は、stcmd lck を使用して、ルート フォルダ StarDraw(StarDraw プロジェクトの StarDraw ビュー内)の子である SourceCode 内のすべてのファイルおよび SourceCode の子フォルダ内のすべてのファイルをロック解除します。

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

関連参照

クライアントのコマンドライン操作

サーバーのロックとロック解除: stcmd server-mode

適切なアクセス権または特権を持っていれば、stcmd server-mode を使用してサーバー構成をロックおよびロック解除することができます。サーバー構成をロックすると、バックアップや他の作業を行うときにサーバーへのアクセスを制限できます。サーバーをロックすると、サーバー管理コマンドしか受け付けられなくなります。サーバー構成をロック解除すると通常操作を続けることができます。

注: ユーザー名が「StarTeam」の場合、パスワードが入力済み、あるいは空のパスワードを使用していても、ユーザーのパスワードが要求されます。

構文

このコマンドの構文は次のとおりです。

```
stcmd server-mode [-pwdfile "filePath"] [-cmp] [-encrypt encryptionType] [-q] [-x]
[-stop] -s "serverName" -mode [lock | exlock | unlock]
```

オプション	説明
exlock	サーバーを排他的にロックします。その結果、他のユーザーは誰もアクセスできなくなります。
ロック (lock)	サーバーを非排他的にロックします。管理コマンドだけが実行できます。
-mode	サーバーを排他的にロックするか、非排他的にロックをするかを指定します。 -mode lock を使用すると、サーバーがロックされている間は、サーバー管理コマンドしか受け付けられません。例えば、バックアップ プログラムを実行している間はこのコマンドを使用します。 -mode exlock を使用すると、サーバーのロックを解除するまで、ロックしたユーザーしかサーバーにアクセスできません。例えば、カスタム フィールドを作成している間はこのコマンドを実行します。 -mode unlock を使用すると、再び他のユーザーがサーバーを利用できるようになります。
ロック解除 (unlock)	サーバーをロック解除し、適切なアクセス権を持つすべてのユーザーがサーバーにアクセスできるようにします。
-s	サーバーを指定します。完全な構文は以下のとおりです。 `-s "userName:password@hostName:portNumber"` 例: -s "JMarsh:password@orion:49201" ユーザー名を省略すると、現在のユーザー名が使われます。前記の例では、ユーザー名は「JMarsh」です。 パスワードを省略すると、パスワードの入力を要求されます。前記の例では、パスワードは「password」です。 ホスト名を省略すると、デフォルトで localhost となります。この例では、ホスト名は「orion」です。ポート番号は必須です。この例では、デフォルトのポート番号 49201 を使用しています。

例

以下の例は、stcmd server-mode を使用して Orion 上でポート 1024 を使用しているサーバーをロックします。

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

関連参照

- クライアントのコマンドライン操作

ファイルの削除: stcmd remove

stcmd remove は、バージョン管理からファイルを削除します。指定ファイルとそのリビジョン履歴は、削除する前の時点までプロジェクト ビューを戻さない限り StarTeam では表示されなくなります。

構文

このコマンドの構文は次のとおりです。

```
stcmd remove -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-csf]
[-encrypt encryptionType] [-is] [-q] [-x] [-stop] [-rp "folderPath" | -fp "folderPath"]
[-filter "fileStatus"] [-df] [files...]
```

オプション	説明
-df	ユーザーの作業ファイルを削除します。このオプションを付けないと、作業ファイルはユーザーのワークステーション上の作業フォルダに残ります。
-filter	1 つまたは複数の文字から成る文字列を指定します。各文字が、ファイル ステータスを表します。この文字列には、スペースやその他のホワイトスペースは入れないで下さい。現在、指定されたステータスを持っているファイルだけが削除されます。「ビュー外」のファイルは、バージョン管理からは削除されません。ステータスを表すために使用できる文字は、次のとおりです。C(最新)、M(変更済み)、G(マージ)、O(古いリビジョン)、I(作業ファイルなし)、U(不明)。

例

以下の例は、stcmd remove を使用して、ルート フォルダ StarDraw(StarDraw プロジェクトの StarDraw ビュー内)の子である SourceCode 内のすべての .hm ファイルおよび SourceCode の子フォルダ内のすべてのファイルを削除します。作業ファイルも削除します。

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

関連参照

クライアントのコマンドライン操作

個人用オプションの設定: stcmd set-personal-options

stcmd set-personal-options は、個人用オプションの設定や表示を行うために使用します。現時点では、以下の個人用オプションのみこのコマンドで設定することができます。

- ファイル ステータス情報の保存方法
- ファイル ステータス情報を集中リポジトリの場所に保存するときの保存場所

ファイル ステータス情報は、starteam-client-options.xml ファイルに保存されます。starteam-client-options.xml ファイルがないと、このコマンドはそれをデフォルトの場所に作成します。デフォルトの場所は、-central-status オプション（以下を参照）のデフォルトと同じです。

構文

このコマンドの構文は次のとおりです。

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

オプション	説明
-central-repository	集中リポジトリの場所を指定します。
-central-status	ファイル ステータス情報が、このワークステーション上のこのユーザー用の集中管理場所に保存されます。この場所は、Windows Client や Cross-Platform Client、このコマンドを使用して設定できます（-central-repository オプションを参照してください）。 この場所を設定しないと、デフォルトは以下になります: Windows NT の場合、アプリケーションがインストールされたフォルダ Windows XP の場合、C:¥Documents and Settings¥username¥Local Settings¥ Application Data ¥Borland¥StarTeam¥ Windows 以外のプラットフォームの場合、/user_home_directory/.starteam-client/
-per-folder-status	作業フォルダに、そのフォルダ内のファイルのファイル ステータス情報を含めるように指定します。ファイル ステータス情報は、作業フォルダの子フォルダである .sbas フォルダに保存されます。作業フォルダを移動すると、それと一緒にファイル ステータス情報も移動します。
-list	現在 starteam-client-options.xml に設定されている個人用オプションの一覧を表示します。

例

以下の例は、stcmd set-personal-options を使用してファイル ステータス情報を集中リポジトリの場所に保存することを示します: C:¥JMarsh¥statusinfo。

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

関連参照

クライアントのコマンドライン操作

ファイル ステータスの更新: stcmd update-status

ファイルのステータスを更新すると StarTeam は作業ファイルをチェックアウトしたリビジョンおよびチップバージョンと比較します。たとえば、ファイル リストではファイルが [最新] となっているのに、いまだれかがそのコピーをチェックインしたために実際にはファイルのステータスが [古いリビジョン] になっていることがあります。

ファイル ステータスの更新は、ファイルの更新と同じではありません。ファイルが作業フォルダになければ、ファイル ステータスを更新すると、ファイルのステータスが [作業ファイルなし] であることを知らされますが、ファイルのチェックアウトは行われません。通常、ファイル ステータスの更新はファイルをチェックイン、チェックアウト、追加、または無視するべきか調べるために行います。

例えば、次のような状況が考えられます。

- 「古いリビジョン」、「作業ファイルなし」または「マージ」ステータスを持つファイルをチェックインする。
- 「変更済み」または「マージ」ステータスを持つファイルをチェックアウトする。
- 「ビュー外」ステータスを持つファイルをアプリケーションに追加する。ただし、update-status コマンドでは、「ビュー外」ステータスを持つファイルは、リポジトリに保存されていないので表示されません。

stcmd update-status を使用してファイル名、コマンドを実行する前のステータス、およびコマンドを実行した後のステータスを表示します。次に出力例を示します: x.cpp: status is Current (was Unknown)。

構文

このコマンドの構文は次のとおりです。

```
stcmd update-status -p "projectSpecifier" [-pwdfile "filePath"] [-cmp] [-csf]
[-encrypt encryptionType] [-is] [-q] [-x] [-stop] [-rp "folderPath" | -fp "folderPath"]
[-cfgl "labelName" | -cfgp "stateName" | -cfgd "asOfDate"] [-filter "fileStatus"] [-eol
[on | off ]][-v] [files...]
```

オプション	説明
-cfgd	指定した日付/時刻のものとしてビューを構成します。日時指定の例を以下に示します。 "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 AM PST"
-cfgl	指定されたラベルを使用してビューを構成します。-cfgl、-cfgp、または、-cfgd を指定していなければ、ビューの現在の構成が使用されます。
-cfgp	指定されたプロモーション状態を使用してビューを構成します。
-eol	on に指定すると、行末マーカー(EOL)を使わずにテキストファイルのステータスを計算します。off にすると(デフォルト)、現在の行末マーカー(EOL)のまま、作業ファイルとチップリビジョンに基づいてステータスが更新されます。このオプションを on にすると、LF を行末マーカー(EOL)に持つ作業ファイルを、CR/LF を行末マーカーに持つチップ リビジョンと比較できます。行末マーカーだけが異なる場合は、ステータスが「最新」と見なされます。
-contents	MD5 チェックサムではなくファイルの内容を送信します。
-filter	1 つまたは複数の文字から成る文字列を指定します。各文字が、ファイル ステータスを表します。この文字列には、スペースやその他のホワイトスペースは入れないで下さい。現在、指定されたステータスを持っているファイルだけが更新されます。C(最新)、M(変更済み)、G(マージ)、O(古いリビジョン)、I(作業ファイルなし)、U(不明)。
-v	指定したフォルダの作業フォルダ内のすべてのファイル(「ビュー外」ステータスのファイルを除く)のステータスを表示します。このオプションを指定しないと、ファイル ステータスが変更された場合のみ、ステータスが表示されます。

例

以下の例は、stcmd update-status を使用して SourceCode という名の StarTeam フォルダの作業フォルダ内の各ファイルのステータスが正確か確認します。SourceCode は、ルート フォルダ StarDraw(StarDraw プロジェクトの StarDraw ビュー内)の子です。

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

関連参照

クライアントのコマンドライン操作

VCM コマンドライン ユーティリティ

ビュー比較/マージ コマンドライン ユーティリティ(VCMUtility)では、StarTeam のソース ビューとターゲット ビューを比較し、必要に応じてターゲット ビューに差分をマージすることができます。

このセクションでは、VCMUtility に関連したすべてのヘルプ トピックを提供します。

ビュー比較/マージ は、StarTeam Cross-Platform Client および VCMUtility でのみ使用可能です。

このセクションの内容

VCM コマンドライン ユーティリティの概要(VCMUtility)

VCMUtility はコマンドライン ユーティリティで、StarTeam のソース ビューとターゲット ビューを比較し、必要に応じてターゲット ビューに差分をマージできます。

VCMUtility ドキュメントに含まれるものは以下のとおりです。

- コマンド
- 接続オプション
- セッション オプション
- セッションに保存されないその他のオプション

注: ビュー比較/マージのセッションをコマンドラインで開始し、StarTeam Cross-Platform Client のグラフィカル アプリケーション版ビュー比較/マージで一連のセッションを終了させることもあるでしょう。たとえば、VCMUtility を使って VCM セッションを作成し(おそらく DefaultAction オプションを使用して)、コミットはしないでおきます。この時点で、VCM セッションは自動的に保存されます(必要に応じて、選択した代わりの名前で保存されます)。その後、StarTeam Cross-Platform Client で VCM セッションを開き、確認と必要な調整を行い、それからリポジトリに変更をコミットすることができます。

構文表記規則

コマンドラインの構文は以下の表記規則に従います。

表記	説明
中かっこ {}	必須項目の要素
角かっこ []	省略可能な要素
アングル ブラケット < >	値や値のセットに置き換わる文字またはフレーズを囲みます。たとえば、<file name> は実際のファイル名やパスに、<userid> は実際のユーザー ID に置き換えられます。しかし、多くの場合、アングル ブラケット内の文字やフレーズはもう少し複雑な構文に展開されます。たとえば、<change requests> は CR、CRs、ChangeRequest、ChangeRequests、ChangeRequest *4277 などに置き換えられます。どれを使用するべきか定かでない場合には、このドキュメントの「参照/比較/マージ」セクションにあるトピック「VCMUtility のクイック参照」を参照してください。この「クイック参照」では、<change requests> といった具合に、各フレーズの完全な構文が提供されています。
パイプ |	代替の要素を分割
アスタリスク *	繰り返し指定可能な要素

注: すべてのオプションには大文字と小文字の区別はありません(たとえば、次の文字は同じオプションとして取り扱われます: Server と server)。

VCM ユーティリティのコマンド

VCMUtility [<options file>] [options]

特定の <options file>(第一パラメータとして)、コマンドライン引数、またはその両方でオプションを指定できます。コマンドライン引数は、<options file> に指定したどのオプションよりも優先されます。<options file> では、オプション名が新しい行の 1 文字目から始まる必要があります。オプション名には最初の "-" を含めません。

VCM ユーティリティのオプション ファイル

VCMUtility のオプションは、オプション ファイルとして、VCMUtility コマンドの最初のパラメータに指定することができます。

例:

```
VCMUtility c:¥VCMconfig.txt
```

オプション ファイルでは、1 行につき 1 オプションのみが指定できます。行頭にオプション名を記述し、オプション名の後には少なくとも 1 つのスペースを置きます。オプションに続く値(パラメータ)は複数行にまたがることができます。複数行になる場合、行の先頭にタブを挿入するか、または何も挿入しないで値を続けて指定します。空白行は無視されます。コメントにはダブル スラッシュ(//)を前に付けます。

例:

```
// This is a comment
server jsmith:mypw@somehost:49201
type Rebase
include "/Cygnus/StarTeam/<StarTeam Core>/Server/Common/*.h" +ALL
  *.cpp    *.rc Makefile    // long value continued on a second line

// The line above was blank
save
 my-rebase-session               // value provided on a separate line
```

コマンドライン パラメータ

VCMUtility のオプションは、「-」をオプション名の前に付けることで、コマンドラインで実行することができます。たとえば、Server のオプションをコマンドラインで実行する場合、-server と指定します。オプションに(値などの)パラメータがある場合、パラメータはオプション名に続いて指定します(「-」は不要)。

複合入力ソース

VCMUtility のオプションは、オプション ファイル、コマンドラインのパラメータ、または両方を複合して指定することができます。たとえば、通常使うオプションまたは定型のオプションをオプション ファイルに指定し、その場に応じて実行したい内容をコマンドラインのパラメータとして指定できます。

コマンドラインには、オプション ファイルで使われているものと同じオプションを使うことができます。同じコマンドがオプション ファイルとコマンドラインで使われる場合、コマンドラインのオプションはオプション ファイルのオプションを上書きします。たとえば、オプション ファイルに Source View1 があり、コマンドラインに -Source View2 があると、View2 がソース ビューとして使用されます。

オプション内の Unicode 文字

コマンドラインでパラメータとして渡されるオプションの値のエンコードは実行環境に依存します(例:コマンド シェル)。このため、必要とするオプションの値が実行環境の制約により VCMUtility に渡せない場合、オプション ファイルによりそれらのオプションを渡す必要があります。

オプション ファイルに BOM(Byte Order Mark)が指定されてない場合、ファイルを開く際にはシステムのデフォルト文字セットが使用されます(例:Windows の場合は ANSI [Windows-1252]、Linux の場合は UTF-8)。オプション ファイルに BOM がある場合、ファイルは BOM に指定されたエンコードで解釈されます。UTF-8 または UTF-16 エンコードを使用すれば、すべての Unicode 文字がオプション ファイルで使えます。

(参考)BOM の値:

BOM	エンコード
0xEFBBBF	UTF-8
0xFEFF	UTF-16 BE(ビック エンディアン)
0xFFFE	UTF-16 LE(リトル エンディアン)

真偽値オプション

すべての真偽値(True または False)の値はデフォルトで False です。しかし、オプションに真偽値を指定しない場合、そのオプションは True を指定したとみなされます。したがって、True を指定することなくオプションを宣言するだけで、そのオプションが実行されます。例:

```
// Set these options to True
AutoLogon
BreakLocks
```

省略名

ほとんどのコマンド名とオプション名には、この説明で用いられた「長い名称」に加えて、1 つのまたは複数の「短い名称」(省略名)があります。省略名は VCMUtility コマンドが長いオプションを持つ場合、それを短くするのに役立ちます。省略名は -Help abbreviations コマンドを使うことで閲覧できます。省略名の例:

Help: H または ?。

ActiveProcessItem: ActivePI または API。

SourceLabel: SrcLabel または SL。

ほとんどの場合、この説明にある大文字小文字交じりのスペルで合成された名称(コマンド、パラメータ)は、大文字のみの省略名となります。たとえば、ManualMergeFiles は MMF、AutoMergeProperties は AMP となり、その他も同様です。

終了コード

VCM ユーティリティーは終了コードを返します。その値により実行結果が判断できます。

終了コード	説明
0	エラーなし。
1	致命的エラーあり。
2	部分的に成功。この結果は、比較部分は成功したが、解決できない競合がありコミットに失敗した場合に返されます。

VCM ユーティリティのログ ファイル

VCMUtility は、実行中、情報メッセージ、警告メッセージ、およびエラー メッセージをコンソール ウィンドウ(標準出力)に書き出します。ほとんどの操作で、VCMUtility はその操作を要約したログ ファイルも作成します。コンソール ウィンドウへの出力がそうであるように、ログ ファイルの内容も Verbose オプションを有効にすると、より詳細になります。ログ ファイルは、VCM セッションの新規作成、および Import、Open、Replay、Resume の各コマンドに対して作成されます。ただし、ログ ファイルが開始されるのは、コマンドライン パラメータとオプション ファイル(使用されている場合)の解析で、エラーが検出されなかった場合だけです。ログ ファイルは、Help コマンドおよび Delete コマンドに対しては作成されません。

VCMUtility のログ ファイルは、ユーザーのホーム ディレクトリ(Java は **user.home** で識別)に、次のファイル名で作成されます:

```
VCMUtility-YYYY-MM-DD_hh-mm-ss.log
```

YYYY-MM-DD および hh-mm-ss にはローカルの日付および時刻を使用します。ログ ファイルが開始されると、その完全パス名がコンソール ウィンドウに出力されます。

VCM ユーティリティの変更パッケージのサポート

StarTeam 2009 リリースから、VCMUtility は、2009 リリースにアップグレードされたすべての StarTeam 構成の変更パッケージをサポートします。変更パッケージは、サーバーに保存される永続オブジェクトなので、多くの利点があり、VCM セッション(.vcms)ファイルや VCM エクスポート(.vcmx)ファイルよりも優れています。そのため、アップグレードされた StarTeam 構成では、セッションを保存および再開するのに、セッション ファイルやエクスポート ファイルよりも、変更パッケージを推奨します。同様に、パラメータを指定した Save オプション、Resume コマンド、Export コマンド、および Import コマンドよりも、パラメータを指定しない Save オプションおよび Open コマンドを推奨します。ただし、後方互換性のために、2009 では、VCMUtility は VCM セッション ファイルを使用するコマンドもサポートし続けます。詳細については、Open コマンド、および Save オプションを参照してください。

関連参照

- VCMUtility のコマンド
- VCMUtility の接続オプション
- VCMUtility のセッション オプション
- VCMUtility のその他のオプション
- VCMUtility の使用例
- クイック参照

VCMUtility のコマンド

ここでは、VCMUtility の機能を、コマンドを用いて実行する場合について説明します。VCMUtility を実行するたびに、コマンドが 1 つ実行されます。

構文の表記規則については、関連参照のリンク「VCM コマンドライン ユーティリティの概要」を参照してください。

VCMUtility のコマンド

VCMUtility [<options file>] [options]

特定の <options file>(第一パラメータとして)、コマンドライン引数、またはその両方でオプションを指定できます。コマンドライン引数は、<options file> に指定したどのオプションよりも優先されます。<options file> 内のオプション名は、行の先頭から始める必要がありますが、先頭の "-" は不要です。

VCMUtility のコマンド タイプ

ここでは、VCMUtility のコマンド タイプについて説明します。デフォルトのコマンド タイプは VCM セッションの新規作成です。

セッションを新規作成するコマンド

デフォルトで、VCMUtility の実行は、次のコマンド --- Help、OPEN、Replay、Resume、Delete、Import --- を明示的に与えられない限り、新しい VCM セッションを開始します

Help(ヘルプ)コマンド

-?

Help [<option>]

VCMUtility のヘルプを表示します。<option> を指定すると、そのトピック固有のヘルプが表示されます。たとえば、Help MMF と入力した場合、ManualMergeFiles オプションのヘルプが表示されます。

Delete(削除)コマンド

Delete <VCM session file>

指定した <VCM session file> に保存されているセッションの削除を指定します。すべての中間ファイル(マージ結果ファイルなど)、およびセッション ファイルそのものが削除されます。ただし、セッションが既にターゲット ビューのコミットされていない変更パッケージとして保存されている場合、その変更パッケージ オブジェクトは削除されません。

Import(インポート)コマンド

Import <VCM exchange file>

Import コマンドは、<VCM exchange file> を渡す点以外は、Resume コマンドと同じです。この VCM 交換ファイル(.vcmx)は、あらかじめ Export コマンドで作成しておく必要があります。インポートされた VCM セッションは、中止された時点から再開されます。

- 比較フェーズがまだ正常に完了していない場合は、比較フェーズが実行されます。
- ManualMergeFiles が指定されていて、既にファイル間に競合が存在する場合は、手動マージが実行されます。
- CheckoutPreview が指定されていて、まだコミットが実行されていない場合は、ターゲット ビューの "マージ プレビュー" がチェックアウトされます。
- ReportDiffs が指定されていて、まだコミットが実行されていない場合は、差分レポートが生成されます。

- CommitMerge が True で、まだコミットが実行されていない場合は、コミット フェーズが実行されます。
- ReportUpdates が指定されていて、コミットが完了している場合は、更新レポートが生成されます。

Export と Import を組み合わせて使用すると、VCM セッションを別のワークステーションに "移動" させることができます。たとえば、あるユーザーが VCM セッションを新規作成し、すべての競合を解決してから、そのセッションを Export できます。その後で、作成されたアーカイブ ファイルをテスト環境に移動し、その環境で Import コマンドを CheckoutPreview オプション(CommitMerge を **False** に設定)と組み合わせて使用して、ターゲットのマージ "プレビュー" をチェックアウト、ビルド、およびテストできます。テストが成功したら、テスト環境で Resume コマンドを実行して、CommitMerge を True に設定できます。

注: Resume コマンドまたは Import コマンドにより再開されたセッションは、コミットされない場合、自動的に保存されます。Save オプションを指定した場合、指定した <VCM session file> にそのセッションが保存されます 。それ以外の場合、Resume コマンドで指定された VCM セッション ファイルが使用されます。なお、Import コマンドでは、自動生成された VCM セッション ファイル名が使用されます。

Open(開く)コマンド

Open <Change Package name>

事前に指定された名前で変更パッケージとして保存された、VCM セッションを再開します。このオプションは、変更パッケージをサポートするサーバーでのみ、使用可能です。指定する変更パッケージ名は、デフォルトかユーザーが指定した名前である必要があり、指定した Project および TargetView に属する、保存済みのコミットされていない変更パッケージの名前でなければなりません。また、セッションは他のユーザーにロックされていてはなりません。ロックされている場合、そのユーザーによって既に開かれていることが多いからです。

詳細については、Name、Save、Import、および Resume コマンドを参照してください。

Replay(再生)コマンド

Replay <Change Package name>

新しいターゲット ビューに対して、あらかじめコミットされた変更パッケージを "再生" することで、VCM セッションを新規作成します。このコマンドは、サーバーが変更パッケージをサポートしている場合にのみ、使用可能です。指定する変更パッケージは、Project オプションで指定されるプロジェクトと、SourceView オプションで特定されるビューに属している必要があります。(コミットされた変更パッケージは、これらが更新するターゲット ビューに "属して" いるので、再生される変更パッケージのターゲット ビューは、常に新規セッションのソース ビューになります。)

Replay コマンドを使用するときは、新規セッションの MergeType が 2 つのビューの関係に基づいて自動的に選択されるように、TargetView を指定する必要があります。

- ターゲット ビューがソース ビューの子である場合は、再ベース セッションが実行されます。
- ターゲット ビューがソース ビューの親である場合は、プロモート セッションが実行されます。
- これ以外の場合は、複製セッションが実行されます。

ターゲット ビューが必要ない場合は、MergeType にプロモートを指定することもできます。

再生 VCM セッションは、新しいターゲット ビューで、指定した変更パッケージ内で行われたのと同じ変更を試みます。このことは、新規作成された VCM セッションのソース スコープが自動的に選択されることを意味します。このため、Include および Exclude オプションは指定できません。再生セッションでは、元の変更パッケージで行われた変更の一部が、新しいターゲット ビューでは行えないことがあります(新しいバージョンが既に存在する場合など)。一部の変更は、異なる方法(たとえば、Merge ではなく Move-and-Merge)で適用されることがあります。また、(Repin ではなく Merge など)新たな競合が発生する可能性もあります。再生セッションは、未解決の競合が発生しない場合にのみ、コミットされます。

Resume(再開)コマンド

Resume <VCM session file>

新規セッションを作成する代わりに、指定した <VCM session file> に保存されたセッションを再開することを指定します。通常これは、比較フェーズしか実行されていない、以前のセッションの、コミット フェーズを実行するのに使用されます。既にコミットされたセッションも、再開することができますが、差分レポートが生成されるだけです。詳細については、Export オプション、および Import コマンドを参照してください。

関連参照

VCM コマンドライン ユーティリティの概要(VCMUtility)
VCMUtility の接続オプション
VCMUtility のセッション オプション
VCMUtility のその他のオプション
VCMUtility の使用例
クイック参照

VCMUtility の接続オプション

このセクションでは、VCMUtility の接続オプションについて説明します。

AutoLogon（自動ログオン）

AL

AutoLogon [True] | [False]

<user> が Server オプションに指定されていない場合、AutoLogon は、StarTeam Toolbar Utility に保存されているユーザー ID/パスワードを使用して、指定した StarTeam サーバーに対してログオンを試みるよう要求します。

Encryption（暗号化）

encrypt

En

Encryption {None | RC4 | RC2_ECB | RC2_CBC | RC2_CFB}

指定した暗号レベルをサーバー接続に使用します。デフォルトは None（なし）です。しかし、SDK が必要に応じて VCMUtility の暗号化レベルを StarTeam Server が要求する最低限のレベルまで自動的に引き上げます。

PwdFile（パスワード ファイル）

PF

PwdFile <file name>

ログオンのパスワードを含んだファイルを指定します。Server パラメータの <password> の代わりに、-PwdFile が用いられます。

Server（サーバー）

ss

Server [<user>[:<password>]@]<host>[:<port>]

VCMUtility が接続する StarTeam Server を指定します。

- <user> および AutoLogon が指定されていない場合、<user> ユーザーは、デフォルトの "Administrator" となります。
- <password> および PwdFile を指定しない場合、VCMUtility がパスワード入力を要求します。
- <user> または <password> に ":"、"@"、または空白文字（ブランク）を使用する場合は、それらのパラメータを一重引用符（'）または二重引用符（"）で囲む必要があります。
- <user> または <password> が引用符で囲まれている場合、バックスラッシュ（エスケープ文字）（¥）を前に付けることで、引用符そのものを名前に含めることができます。
- 引用符で囲まれた <user> または <password> にバックスラッシュ（エスケープ文字）が含まれている場合、直前に別のバックスラッシュを置きます。たとえば、2 つ続いたバックスラッシュは 1 つのバックスラッシュとして認識されます。
- <host> にはホスト名または IP アドレスを指定します。<host> は Server コマンドのオプションを宣言するときの必須パラメータです。

- Server コマンドのオプションを指定しない場合、<host> はデフォルトの **localhost** になります。指定されない場合の <port> はデフォルトの 49201 になります。

UseCA（Cache Agent の使用）

UCA

UseCA {<host>:<port> | AutoLocate}

ファイルのチェックアウト時に MPX Cache Agent の使用を試みます。Cache Agent を明示的に指定する場合、ホスト名またはアドレス（<host> とポート番号 <port>）を指定します。ネットワークで最も近い Cache Agent を使用する場合は AutoLocate を指定します。

UseServerProfile（サーバー プロファイルの使用）

USP

UseServerProfile [True | False]

[True] の場合、Server のオプションで指定されている <host> の名前が、サーバー プロファイル名として解釈されます。サーバー プロファイルは、ユーザーの starteam-servers.xml ファイルに保存されています。サーバー プロファイルには、StarTeam Server のホスト名、ポート番号、暗号化レベル、比較設定などが指定されています。このため、UseServerProfile を指定する場合には、Server オプションを指定する必要がありますが、それにポート番号をつけてはいけません。また、Encryption オプションも指定してはいけません。

関連参照

VCM コマンドライン ユーティリティの概要（VCMUtility）
VCMUtility のコマンド
VCMUtility のセッション オプション
VCMUtility のその他のオプション
VCMUtility の使用例
クイック参照

VCMUtility のセッション オプション

このセクションでは、新規セッションで利用できる VCMUtility オプションについて説明します。

説明は 2 つのセクション（"新規セッションのオプション" と "再開セッションのオプション"）に分かれています。

新規セッションのオプション

このセクションでは VCMUtility セッション オプションについて説明します。

AutoMergeFiles（自動ファイル マージ）

AMF

AutoMergeFiles [True | False]

True の場合、比較時にステータスがマージになったファイルに対して、自動マージを実行します。自動マージが成功した場合、マージされたファイルが VCM セッションに保存されます。失敗した場合、マージされたファイルは破棄され、ファイルのマージ ステータスは未解決のままです。AutoMergeFiles は比較セッションでは無視されます。

AutoMergeProperties（自動プロパティ マージ）

AMP

AutoMergeProperties [True | False]

True の場合、比較時にステータスがマージのアイテムに対して、プロパティの自動マージを実行します。自動マージが成功した場合、マージされたアイテムが VCM セッションに保存されます。失敗した場合、マージされたアイテムは破棄され、アイテムのマージ ステータスは未解決のままです。AutoMergeProperties は比較セッションでは無視されます。

BreakLocks（ロック解除）

BL

BreakLocks [True | False]

True の場合、比較時にロックされているアイテムが見つかると、ロック解除を試みます。ソースまたはターゲット アイテムに対してロックの解除が必要なケースは、別のユーザーがそのアイテムをロックしている場合に限ります。ロックの解除には特別の権限が必要であり、権限により失敗することもあります。BreakLocks は比較セッションでは無視されます。

CaseSensitiveFilenames（ファイル名の大文字と小文字を区別する）

CSF

CaseSensitiveFilenames [True | False]

True の場合、PreventDuplicateFilenames オプションの評価に大文字と小文字のみが異なる同名のファイルを別のファイルとして認識させます。また、ソースとターゲット ビューのファイルを比較する時にも使われます。

CheckoutPreview（チェックアウト プレビュー）

Checkout

CP

CheckoutPreview <files> [<check-out options>]

このオプションでは、クライアントのワークスペースにチェックアウトする "マージ プレビュー" 内のファイルを指定します。"マージ プレビュー" とは VCM セッションですべての変更が更新されたターゲット ビュー（シミュレーション）です。
<files> 構文ではマージ プレビューに特定フォルダからチェックアウトするファイル名やパターンを指定できます。任意指

定の <check-out options> では(ファイルの)チェックアウト先のフォルダやチェックアウトするファイル ステータスの条件が指定できます。

CheckoutPreview を指定すると、ファイルが比較されたのち、自動マージまたは手動マージ後にチェックアウトされます。ただし、これらのファイルはコミットされていません。チェックアウトは、VCM セッションのファイル マージに競合がなかった場合に限り行われます。マージで競合があった場合はエラーが表示され、CommitMerge オプションにかかわらず、マージされません。競合がなく、かつ、CommitMerge で True が指定されている場合、VCM セッションはファイルをチェックアウト後にコミットします。

例:

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

CommitMerge(マージ結果をコミット)

Commit

CM

CommitMerge [True | False]

VCM セッションの結果をコミットするかどうかを指定します。パラメータに False を指定した場合、コミットされません。この場合、比較/レポートのみを実行するセッションが作成できます。True を指定すると、解決できない競合がある場合を除いてコミットします。CommitMerge は比較セッションでは無視されます。

DefaultAction(デフォルト アクション)

DA

DefaultAction [MergeType <merge type>] [ItemType <item type>] <match state> <action>

比較対象アイテムに対して、デフォルトの <action>(アクション)を指定します。アクションの実行条件は、<match state>(一致条件)に指定します。ソース ビューとターゲット ビューのアイテムに差分がある場合、VCM ユーティリティは実行するアクションの種類を決定するためのルールからなる "決定表" を使用します。DefaultAction オプションを使用することにより、(ルールに定められた)デフォルトのアクションを上書きできます。複数の差分に対して、このオプションを複数回宣言できます。ただし、アクションの定義順序は重要です。比較時に発見された差分に対して、条件の合うアクションが複数見つかった場合、一番最後のアクションのみが実行されます。

- MergeType を使用する場合、DefaultAction に適用可能な <merge type> は、Rebase(再ベース)、Promote(プロモート)または Replicate(複製)です。 は、Rebase(再ベース)、Promote(プロモート)または Replicate(複製)です。
- MergeType を指定しない場合、DefaultAction が現在の VCM セッションに適用されます。

DefaultAction に現在のセッションで使われない <merge type> を指定した場合、そのオプション ファイル内で実行される別の VCM セッションに指定したルールが適用されます。

MergeType を使用する場合、DefaultAction に適用可能な <merge type> は、Rebase(再ベース)、Promote(プロモート)または Replicate(複製)です。 は、CRs(変更要求)、Files(ファイル)、Folders(フォルダ)、Requirements(要件)、Tasks(タスク)、Topics(トピック)です。デフォルトでは DefaultAction はすべてのタイプに適用されます。

<match state> は条件を指定するパラメータです。比較中、ソースやターゲットが、ここで指定した条件と一致するのか照合します。<match state> にはソース、ターゲットの <item condition>(アイテム コンディション)の一致条件を指定します(複数指定可能)。<item condition>には <condition name>(コンディション名、source.moved など)および <condition value>(コンディションの値、True、False または Unspecified)を指定します。<condition value> は任意指定のパラメータで、デフォルトの値は True です。<match state> は、すべてのコンディションを含めることができます。

<action> は差分が <match state> で指定した条件を満たした場合、どのようにソース、ターゲットのアイテムが処理されるかを指定します。<action> には、単に条件を満たしたアイテムのデフォルトのアクションを定義します。実際のアクションは StarTeam Cross-Platform Client で比較が行われた後に変更できます。

DefaultAction の定義例を次に示します。

```
//When a source item has moved, but the target item has not,
//ignore the move.
DefaultAction source.moved target.moved false Ignore

//In a Rebase, if a file is binary and has been modified in both the
//source and target, overwrite the target with the source version.
DefaultAction MergeType Rebase
   items.binaryfile
   source.modified
   target.modified
   Overwrite

//In a Promote, if a CR has moved in both the source and target views
//(to different folders), move the target item to the matching folder as
//the source item, but only if the CRs are on the same branch.
DefaultAction MergeType Promote ItemType CR
   source.moved
   target.moved
   items.branched false
   Move
```

DefaultAction は比較セッションでは無視されます。

DefaultComment（デフォルトのコメント）

DC

DefaultComment <comment>

デフォルトのリビジョン コメントを指定します。このコメントは新しいアイテム（リビジョン）がターゲット ビューで作成されたときに使用されます。<comment> には任意のテキスト文字列を指定します。改行文字（CR と LF）、空白文字、およびタブを含め、コメントに存在する連続したスペースは、それぞれの場所で 1 文字の空白文字に変換されます。デフォルトでは、新しいアイテムのリビジョンに対して、自動生成コメント（デフォルト リビジョン コメント）が使用されます。デフォルトのリビジョン コメントを無効にするには、DefaultComment オプションのみ指定します（<comment>なし）。

DefaultComment は比較セッションでは無視されます。

Exclude（除外）

Exc

Exclude <folders>

ソース スコープから除外するフォルダを指定します。Include <files> または Include <folders> に明示的に指定されたフォルダのみ除外されます。このため、Exclude <folders> オプションは、ソース スコープから不必要なフォルダを除外するために使われます。

使用例：

```
//Include CRs and files in all folders below /a/b/
Include /a/b/ +all CRs Files

//But exclude CRs in folder /a/b/c/
Exclude /a/b/c/ CRs
```

```
//But if this CR is in folder /a/b/c/, it is still included
Include CR 12345
```

オプションの宣言順序に関わりなく、Exclude オプションは Include オプションの後に処理されます。

Include と Exclude の構文

Include が宣言されない場合、デフォルトの VCM セッション スコープは、暗黙的に "ソース ビューにあるすべてのファイル" になります。同じことを明示的に宣言すると include /* +all になります。Include オプションを使用すると、スコープがセッション全体を通じて Include 宣言文で指定したアイテムのみに限定されます。暗黙的なスコープを用いる場合、またはスコープを明示的に宣言する場合どちらも、Exclude オプションによりソース アイテムを除外できます。

すべての Include および Exclude オプションはソース ビューのオブジェクト(ラベル、ファイル、変更要求など)を指定する必要があります。ここでも、タイプ名には単数形、複数形どちらも使用できます(RevLabel、CR など)。複数のアイテムを対象にする場合においても単数形が使用できます。

注: Exclude オプションは宣言順序に関わりなく、Include オプションの後に処理されます。したがって、Exclude /src/foo/bar/ が Include /src/foo/ +depth all の前に宣言される場合も /src/foo/bar/ はスコープから除外されます。

Export(エクスポート)

Exp

Export <VCM exchange file>

Export オプションを宣言すると、VCM セッションはマージ結果を含む情報を結合し、<VCM exchange file> で指定するファイルに保存します。ファイル名(Exchange File)には、常に拡張子 .vcmx が付きます。VCM 交換ファイルでは VCM セッション全体が別のマシンに転送できます。これによりセッションを再開する Import コマンドを別のマシンで実行できます。詳細は Import コマンドのセクションを参照してください。

<VCM exchange file> にパスが含まれない場合、ファイルはユーザーのホーム ディレクトリ(Java の user.home)に保存されます。

注: Export オプションは、セッション自体が保存されないときでも、常に VCM Exchange File を作成します。詳細は Save コマンドのセクションを参照してください。

FixFloatingChildShares(子の変動共有を固定)

FFCS

FixFloatingChildShares [True | False]

再ベースと複製のマージにおいて、ソース ビューに変動共有しているターゲットのアイテムを固定(ピン)することにより "修正" します。ターゲット ビューのアイテムがソース アイテムの変動共有の子の場合(ターゲットのアイテムが分岐されていないという意味)、共有によりソース アイテムに対する変更が即座にターゲットに反映されるために、VCM のセッション中にソースとターゲットのアイテムの間に差分が見つかることはありません。VCM のベスト プラクティスに基づき、変動共有のアイテムは常に固定するという方法を推奨します。これにより、ソースでの変更がターゲット ビューに管理された状態で反映されるようになります。このオプションにより変動共有の子のアイテムを親アイテムのリビジョンに固定する "修正" をすることができます。このオプションの利用は、比較時に各ターゲット アイテムに変動共有が設定されているかどうかを調べるため、パフォーマンスが低下します。

IgnoreMergePoints(マージ ポイントを無視)

IMP

IgnoreMergePoints [True] | [False]

比較時にマージ ポイントを無視するかどうかを指定します。True である場合、マージで競合の発見されたアイテムは、前回のマージ ポイントのソース リビジョンの代わりに、共通の祖先を分岐ポイントとして使います。

Include(インクルード)

Inc

Include {<change requests> | <files> | <folders> | <process items> | <requirements> | <revision labels | <tasks> | <topics> }

ソース スコープに指定したアイテムを含めます。Include オプションは何度でも使用でき、それらすべてをソース スコープに含めることができます。1 つの Include オプションに対して、1 つのアイテム タイプ(リビジョン ラベル、変更要求など)のみが指定できます。タイプのキーワード(ファイルとフォルダの場合はオプション)には単数形、複数形ともに使用できます。たとえば、ProcessItem または ProcessItems です。

例:

```
Include CRs ALL
Include /src/com/*.java +all *.jar +2 *.jpx 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(マージの競合をロック)

LMC

LockMergeConflicts {None | Source | Target | Both}

未解決の競合があるアイテムがソース ビュー、ターゲット ビューあるいは両方のビューで排他的にロックすることを指定します。比較時にロックが実行されます。None がデフォルトです。未解決ロックのあるアイテムに対してロックを作成できないことを指定します。ロックは差分が発見されたアイテム(ソースまたはターゲット、あるいは両方)にのみ適用されますので注意してください。差分が見つからない比較対象に対してロックは適用されません。このオプションは、プロジェクトのオプションである **ファイルのチェックイン時に排他的ロックが必要** や StarTeam クライアントのオプション **チェックアウト時にファイルを排他的にロックする** とは無関係に動作します。これらのオプションは VCM エンジンによって正しく処理されます。LockMergeConflicts は比較セッションでは無視されます。

ManualMergeFiles(手動ファイル マージ)

MMF

ManualMergeFiles [True | False]

True の場合、マージが実施されるソース、ターゲット ファイルが発見されると、ワークステーションに構成されたマージ ツールが起動されます。

ManualMergeFiles は AutoMergeFiles と組み合わせて使用できます。

- 競合が検出され、AutoMergeFiles が必要な場合、自動マージが先に実行されます。
- 競合が解決された場合、マージ後のファイルは保存され、手動マージは不要となります。
- 自動マージが失敗した場合、または AutoMergeFiles が宣言されていない場合で、ManualMergeFiles が True の場合、手動マージが実行されます。

注: ManualMergeFiles は、ワークステーションに手動マージ ツールが構成されていない場合、警告が表示されて無視されます。手動マージ ツールが起動できない場合やツールからエラーが返された場合、ファイルのマージ ステータスは未解決のままです。ManualMergeFiles は比較セッションでは無視されます。

Match(一致)

Match [Folder] *{<folder path> to <folder path>}

比較目的で指定します。1 番目の <folder path> に指定したフォルダ(ソース ビューに必要)は 2 番目の <folder path> (ターゲット ビューに必要)に一致する必要があります。Match オプションは "あいまい一致" 条件を避けるため場合により必要となります。あいまい一致は一方のビューが非派生ビューであるときに発生することがあります。一般的に、Match オプションはソース ビューとターゲット ビューのルート フォルダが一致するためにのみ必要です。ただし、比較フェーズで報告される他のあいまい一致条件を解決するために他のフォルダを一致させることもできます。

ソースとターゲットの <folder path> の両方で先頭と末尾がフォワード スラッシュ("/")である必要があります。

慣例により、ルート フォルダは単一の "/" で表します。つまり、ルート フォルダ名はフォルダ パスで指定しないということです。たとえば、ルート フォルダ名が "StarDraw" である場合、直下の子フォルダ "Source Code" に対するフォルダ パスは単に "/Source Code/" です。

例:

```
// Force the source and target root view folders to match.
Match / to /
```

```
//Force the source view folder "/Source Code" to match the target view
//folder "/Modules/Materials/src".
Match "/Source Code/" to "/Modules/Materials/src/"
```

MergeType(マージ タイプ)

Type

MT

MergeType {Compare | Rebase | Promote | Replicate}

比較セッションまたは再ベース、プロモート、複製マージ セッションを実行するかどうかを指定します。SourceView のみが指定されている場合、MergeType のデフォルトは Promote(プロモート)です。TargetView のみが指定されている場合、MergeType のデフォルトは Rebase(再ベース)です。SourceView と TargetView の両方が指定されている場合、MergeType を指定する必要があります。比較セッションの場合、ソースとターゲット ビューは同じです。

Name(名前)

Na

Name <Change Package name>

VCM セッションに関連付ける変更パッケージの名前を指定します。変更パッケージをサポートするサーバーでは、セッションの保存またはコミットによって変更パッケージが作成されるときに、自動的に名前が選択されます。このオプションを使用すると、デフォルトの名前の代わりに、指定した名前を使用できます。ただし、指定する名前は、ターゲット ビューで既に保存またはコミットされている他のすべての変更パッケージ名から、一意に区別できなければなりません。区別できない名前の場合、保存アクションまたはコミット アクションは失敗します。

Name オプションを Open コマンドと組み合わせて使用すると、開かれた変更パッケージの名前が、指定された値に変更されます。

Save オプション、および CommitMerge オプションについても参照してください。

PostCommitLabel(コミット後のラベル)

PostCL

PostCommitLabel <label>

VCM セッションがコミットされる場合、すべての更新が完了した後、ターゲット ビューに対象**ビュー** <label>(ラベル)が作成されます。ラベルは比較時に使われたターゲット ビューのすべてのアイテム(コミット時に**変更された**もの)のリビジョンを反映します。したがって、出来上がったラベルは新しいアイテム、新しいアイテム リビジョン、およびアイテムの移動を含むことになります。しかし、コミットで削除されたアイテムはラベルから除外されます。コミット後のラベルは実質的に直前の "マージ プレビュー" と一致します。PostCommitLabel は比較セッションでは無視されます。

デフォルトでは、ポストコミット ビュー ラベルはデフォルト名で作成されます。ポストコミット ビュー ラベルを無効にするには、PostCommitLabel に空白値(" ")を指定します。

PostCommitRevLabel(コミット後のリビジョン ラベル)

PostRL

PostCommitRevLabel <label>

VCM セッションがコミットされた場合、指定した**リビジョン**の <label>(ラベル)がターゲット ビューに作成されます。削除されたアイテムを除き、VCM セッションで変更されたファイルすべてにラベルが付きます。この結果、ラベルは VCM セッションにより追加、移動、再固定、更新されたアイテムを含むことになります(削除されたアイテムは除く)。
PostCommitRevLabel は比較セッションでは無視されます。

デフォルトでは、ポストコミット リビジョン ラベルは作成されません。

PreCommitLabel(コミット前のラベル)

PreCL

PreCommitLabel <label>

ターゲット ビューにビュー <label>(ラベル)を作成します。このラベルには比較時に使われたスナップショットが反映されます。ラベルは、比較時に使われたすべてのターゲット ビューのアイテムのリビジョンに反映されます。PreCommitLabel は比較セッションでは無視されます。

デフォルトでは、プリコミット リビジョン ラベルは作成されません。

PreCommitRevLabel(コミット前のリビジョン ラベル)

PreRL

PreCommitRevLabel <label>

VCM セッションをコミットした場合、指定したリビジョン <label>(ラベル)がターゲット ビューに作成されます。ターゲット ビューにある無視されないすべてのアイテムに、"過去の" ステータスとしてこのラベルが付きます。これにより、セッションによって変更される前に、ターゲット ビューのアイテムにリビジョン ラベルが付けられます。つまり、ターゲット ビューに追加されるアイテムにはラベルが**付きません**が、削除されるアイテムにはラベルが**付きます**。PreCommitRevLabel は比較セッションでは無視されます。

デフォルトでは、プリコミット リビジョン ラベルは作成されません。

PreventDuplicateFilenames(重複ファイル名を回避)

PDF

PreventDuplicateFilenames [True | False]

True の場合、マージの結果、同じフォルダに 2 つの同名のファイルができる結果になると、ターゲット ビューに新しいファイルを共有することが許可されません。

Project(プロジェクト)

Pro

Project <project>

VCM セッションで使用するプロジェクトを指定します。このオプションは必須です。ソース ビューとターゲット ビューは同じ <project> に所属している必要があります。プロジェクト名には大文字と小文字の区別はありません。

ReportDiffs(差分をレポート)

RD

ReportDiffs [True | False]

True の場合、比較時に発見されたアイテムの差分を一覧出力します。差分レポートはユーザーのホーム ディレクトリに保存されます(Java の user.home)。差分レポートのタイトルは、

```
VCMDiffReport-YYYY-MM-DD_hh-mm-ss.html
```

YYYY-MM-DD および hh-mm-ss にはローカルの日付および時刻を使用します。

ReportUpdates(更新をレポート)

RU

ReportUpdates [True | False]

True の場合、コミット時に発生したターゲット ビューに対するすべての変更を一覧出力します。更新レポートはユーザーのホーム ディレクトリに保存されます(Java の user.home)。更新レポートのタイトルは、

```
VCMUpdateReport-YYYY-MM-DD_hh-mm-ss.html
```

YYYY-MM-DD および hh-mm-ss にはローカルの日付および時刻を使用します。

ReportUpdates は比較セッションでは無視されます。

Save(保存)

Save [<VCM session file>]

VCM セッションを保存することを指定します。デフォルトでは、コミットされていない VCM セッションは、以下の形式のデフォルトの名前で、VCM セッション ファイル(.vcms)に自動的に保存されます。

```
<user home>/VCMSession-YYYY-MM-DD_hh-mm-ss.vcms
```

YYYY-MM-DD_hh-mm-ss はセッションが保存されたときの日時です。フォルダ <user home> はユーザーのホーム ディレクトリです。

Save オプションに <VCM session file> 名を指定すると、コミットされていないセッションを、デフォルトのファイル名ではなく、指定したファイル名で保存できます。必要な場合は、.vcms を名前に追加します。ファイル名にパスを指定しないと、セッション ファイルは user.home フォルダに保存されます。

.vcms ファイルには VCM セッションのメタデータが含まれますが、マージされたファイルの内容は含まれません。マージされたファイルの内容はユーザー指定の一時フォルダに保存され、セッション ファイルの要素から参照されます。このため .vcms ファイルは、同じワークステーション上でその VCM セッションを再開するのに、使用できるだけです。Resume コマンドを参照してください。

ファイル名を指定しないで、Save オプションを指定すると、コミットされていない VCM セッションを、ターゲット ビューのアクティブな変更パッケージとして保存する試みが実行されます。この変更パッケージは、デフォルトの名前か、ユーザーが指定した名前(Name オプション参照)で、保存されます。変更パッケージとして保存された VCM セッションは、後で Open オプ

ションを使用して、どのワークステーションでも再開できます。ただし、サーバーが変更パッケージをサポートしていない場合、またはサーバー側での保存が失敗した場合は、そのセッションは上述のデフォルトのファイル名で、変更パッケージではなく .vcms ファイルに保存されます。

コミットが正常に実行された場合は、Save オプションは無視されます。サーバーが変更パッケージをサポートしている場合は、コミットされたセッションが、デフォルトの名前かユーザーが指定した名前(Name オプション参照)を使用して、コミット済み変更パッケージを作成します。.vcms ファイルが既に作成されている場合は、その VCM セッションで作成されたすべてのマージ結果ファイルと共に、そのファイルも削除されます。

Export オプションも参照してください。

SourceLabel(ソース ラベル)

SrcLabel

SL

SourceLabel <label>

ソース ビューが該当ビュー ラベルとして使用されることを要求します。ラベル名には大文字と小文字の区別はありません。SourceLabel、SourceState と SourceTime のいずれか 1 つのみ指定できます。これらのオプションのいずれも指定されていない場合、SourceTime Now オプションが暗黙的に使用されます。

SourceState(ソース状態)

SrcState

SS

SourceState <state>

ソース ビューが該当ビューのプロモーション状態として使用されることを要求します。プロモーション状態の名前では、大文字と小文字が区別されません。SourceLabel、SourceState と SourceTime のいずれか 1 つのみ指定できます。これらのオプションのいずれも指定されていない場合、SourceTime Now オプションが暗黙的に使用されます。

SourceTime(ソース時刻)

SrcTime

ST

SourceTime {<timestamp> | Now}

ソース ビューが該当タイムスタンプとして使用されることを要求します。キーワード Now は現在の時刻のスナップショットが構成タイムスタンプとして使用されるようにします。SourceLabel、SourceState と SourceTime のいずれか 1 つのみ指定できます。これらのオプションのいずれも指定されていない場合、SourceTime Now オプションが暗黙的に使用されます。

SourceView(ソース ビュー)

Source

SV

SourceView <view>

VCM セッションで使用するソース ビューを指定します。プロジェクトに同名の <view> が複数存在する場合、スラッシュで区切られたビューのパスを指定します(たとえば MainView/ChildView/GrandchildView)。ビュー名にスラッシュが用いられている場合は、引用符でビュー名を囲みます。

SourceView は Rebase のマージに対しては任意設定事項です。指定した場合、ターゲット ビューにはソース ビューの親ビューを指定する必要があります。

注: ビュー名には大文字と小文字の区別はありません。

TargetLabel(ターゲット ラベル)

TgtLabel

TL

TargetLabel <label>

ターゲット ビューが該当ビュー ラベルとして使用されることを要求します。TargetLabel は比較セッションにのみ使用できます。ラベル名には大文字と小文字の区別はありません。TargetLabel、TargetState と TargetTime のいずれか 1 つのみ指定できます。これらのオプションのいずれも指定されていない場合、TargetTime Now オプションが暗黙的に使用されます。

TargetState(ターゲット状態)

TgtState

TS

TargetState <state>

ターゲット ビューが該当ビューのプロモーション状態として使用されることを要求します。TargetState は比較セッションにのみ使用できます。プロモーション状態の名前では、大文字と小文字が区別されません。TargetLabel、TargetState と TargetTime のいずれか 1 つのみ指定できます。これらのオプションのいずれも指定されていない場合、TargetTime Now オプションが暗黙的に使用されます。

TargetTime(ターゲット時刻)

TgtTime

TT

TargetTime {<timestamp> | Now}

ターゲット ビューが該当タイムスタンプとして使用されることを要求します。TargetTime は比較セッションにのみ使用できます。キーワード Now は現在の時刻のスナップショットが構成タイムスタンプとして使用されるようにします。TargetLabel、TargetState と TargetTime のいずれか 1 つのみ指定できます。これらのオプションのいずれも指定されていない場合、TargetTime Now オプションが暗黙的に使用されます。

TargetView(ターゲット ビュー)

Target

TV

TargetView <view>

VCM セッションで使用するターゲット ビューを指定します。プロジェクトに同名の <view> が複数存在する場合、スラッシュで区切られたビューのパスを指定します(たとえば MainView/ChildView/GrandchildView)。ビュー名にスラッシュが用いられている場合は、引用符でビュー名を囲みます。

TargetView は Promote のマージに対しては任意設定事項です。指定した場合、ソース ビューにはターゲット ビューの親ビューを指定する必要があります。比較セッションでは、ターゲット ビューはソース ビューと同じにできます。

注: ビュー名には大文字と小文字の区別はありません。

再開セッションのオプション

新規セッションに使用できるすべてのオプションは、再開セッションにも指定できるため、Resume コマンドに対して同じオプション ファイルを指定できます。ただし、ほとんどのオプションはセッション開始後には変更できないので、再指定しても無視されます。以下に例外のオプションを示します。

- 接続オプション: 接続情報(サーバー アドレスとポート番号、ユーザー ID とパスワード)は VCM セッション ファイルに保存されないため、再開セッションで再指定する必要があります。ただし、同じ StarTeam Server に再接続しない場合や、VCM セッションのビューやアイテムに対して必要な権限がない異なるユーザーを使用する場合、再開セッションは失敗します。
- CommitMerge: 通常、再開セッションでは、このオプションを True に指定します。これにより、最初の VCM ユーティリティ実行を比較のみ実行に使用し、2 回目の VCM ユーティリティ実行をコミット実行に使用できます。
- ReportDiffs: このオプションは再開セッションで指定できます。True に設定すると、出力すべき内容がある場合、コミット前に差分レポートが作成されます。
- ReportUpdates: このオプションは再開セッションで指定できます。True に設定して、コミットが成功すると、ターゲット ビューに対して行われたすべての変更がレポート出力されます。
- CheckoutPreview: 通常、元の VCM セッションに CheckoutPreview が指定されていると、再開セッションでは、以前と同じオプションを指定した "マージ プレビュー" のチェックアウト操作が実行されます。ただし、再開セッションに CheckoutPreview を指定すると、元のオプションはオーバーライドされ、再開セッションでは、新しい設定に応じてファイルがチェックアウトされます。
- Description: このオプションを指定すると、デフォルトまたは事前に設定された、変更パッケージの説明をオーバーライドできます。新しい説明は、VCM セッションを保存するかコミットしたときに作成される、新しい変更パッケージのリビジョンに使用されます。
- ManualMergeFiles: 通常、元の VCM セッションに ManualMergeFiles が指定されていて、そのセッションがファイル マージの競合が未解決の状態で保存されると、セッションが再開されたときに手動マージ フェーズが再度実行されます。ただし、元の VCM セッションに ManualMergeFiles が指定されていない場合、再開セッションでこのオプションを True に指定すると、手動マージ フェーズを開始できます。また、再開セッションでこのオプションを False に指定すると、手動マージ フェーズを回避できます。
- PostCommitLabel、PostCommitRevLabel、PreCommitLabel および PreCommitRevLabel: これらのラベル オプションを再開オプションに指定すると、それぞれのラベルの前の値がオーバーライドされます。ラベル オプションに空白(" ")を設定すると、対応するラベル オプションが無効になり、コミット フェーズで作成されなくなります。

関連参照

VCM コマンドライン ユーティリティの概要(VCMUtility)
VCMUtility のコマンド
VCMUtility の接続オプション
VCMUtility のその他のオプション
VCMUtility の使用例
クイック参照

VCMUtility のその他のオプション

このセクションでは、ビュー比較/マージ セッションには保存されない VCMUtility のその他のオプションを説明します。

NetMon(ネット モニター)

NM

NetMon [True | False]

SDK のネット モニター機能の使用を可能にします。VCMUtility が StarTeam へ発行した各コマンドは、コンソール ウィンドウにログ出力されます(ただし、VCMUtility ログ ファイルには出力されません)。

Time(時間)

t

Time [True | False]

それぞれの VCM セッションが実行された経過時間を表示します。経過情報は、コンソール ウィンドウと VCMUtility ログ ファイルの両方に出力されます。

Verbose(詳細情報)

vb

V

Verbose [True | False]

実行中、補足的な診断情報や進捗情報がコンソール(標準出力)、および VCMUtiliry ログ ファイルに表示されます。

関連参照

- VCM コマンドライン ユーティリティの概要(VCMUtility)
- VCMUtility のコマンド
- VCMUtility の接続オプション
- VCMUtility のセッション オプション
- VCMUtility の使用例
- クイック参照

VCMUtility の使用例

ここでは、VCMUtility を用いた各種マージ タイプの例を示します。

Hello World Rebase(再ベースの Hello World)

VCMUtility の再ベースを用いて、"Hello World" を実行します。

```
Type     Rebase
Project     Hello
Target     World
```

Automatic Rebase(自動再ベース)

以下のオプション ファイルは上記の例と同様の再ベースを実行します。しかし、可能であればコミットを行い、結果の詳細なレポートを出力します。

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

// All of these options are set to True:
AutoMergeFiles
BreakLocks
ReportDiffs
ReportUpdates
```

すべてのファイルは内容、プロパティ共に自動マージされます。競合が解決できなかったファイルについては、ソース、ターゲット ビュー共にロックされます。可能であれば、他のメンバーがロックしているファイルもロック解除します。解決できない競合がなければ、セッションはコミットされます。比較時の差分およびコミット時の更新内容の詳細はレポートされます。コミットが成功した場合、VCM セッション一時ファイルはすべて削除されます。

Promote by View Label:Compare Only(ビュー ラベルによるプロモート:比較のみ)

以下のオプションは、ビュー ラベルに登録されているファイルおよび変更要求(CR)に対して、比較のみのプロモートを行います。セッションは指定するファイル名を用いて保存されます。

```
// Connection settings
Server            MyUserid@ProdServer:4000
PwdFile            MyPassword.txt

// Merge type and view configuration
Type            Promote
Project            StarDraw
Source            "Beta Release"
SrcLabel        Build-4.0_142

// Select all files and CRs as source items
include            /* +all
include            / +all CRs
```

```
// Compare-only, report, and save with a specific session filename
CommitMerge        False
save            Build-4.0_142-Promote
ReportDiffs

//Miscellaneous options
AutoMergeFiles        True
AutoMergeProperties    False // leave these as conflicts and merge manually
LockMergeConflicts    Target
```

Promote by View Label(ビュー ラベルによるプロモート:マージ)

VCMUtility で以下のコマンドラインを実行すると、上記の例で保存したセッションを再開してコミットします。ここでは、新しい競合が発見されなかったものとします。

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

関連参照

- VCM コマンドライン ユーティリティの概要(VCMUtility)
- VCMUtility のコマンド
- VCMUtility の接続オプション
- VCMUtility のセッション オプション
- VCMUtility のその他のオプション
- クイック参照

クイック参照

VCMUtility のコマンドライン構文: VCMUtility [<options file>] [*<option>]

<options file>(オプション ファイル)内の、それぞれの <option>(オプション)は、行の先頭(1 文字目)から始める必要があります。ただし、行の先頭にスペースまたはタブを挿入すると、前の行に続く行として続けることができます。コマンドラインでオプションを入力する際、各 <option> には、前に "-" を付ける必要があります。

オプション

以下の表示に、VCMUtility のコマンドライン オプションとその構文を示します。

コマンド/オプション

```
<option>
  <command> | <connection option> | <session option> | <miscellaneous option>
```

```
<command>
  {{Help | H | ?} [<help topic>]} |
  {Delete <VCM session file>}|
  {Import <VCM archive file} |
  {Open <Change Package name>} |
  {Replay <Change Package name>} |
  {Resume <VCM session file>}
```

```
<connection option>
  {{AutoLogon | AL} [True | False]} |
  {{Encryption | Encrypt | En} {None | RC4 | RC2_ECB | RC2_CBC | RC2_CFB}} |
  {{PwdFile  | PF} <file name>} |
  {{Server | S} [<user>[:<password>]@]<host>[:<port>]} |
  {{UseCA | UCA} {<host>:<port> | AutoLocate}} |
  {{UseServerProfile | USP} [True | False]}
```

```
<session option>
  {{AutoMergeFiles | AMF} [True | False]} |
  {{AutoMergeProperties | AMP} [True | False]} |
  {{BreakLocks | BL} [True | False]} |
  {{CaseSensitiveFilenames | CSF} [True | False]} |
  {{CheckoutPreview | check-out | CP} <files> [<check-out opotions>]} |
  {{CommitMerge | Commit | CM} [True | False]} |
  {{DefaultAction | DA} [MergeType <merge type>] [ItemType <item type>] <match state> <action>} |
  {{DefaultComment | DC} <comment>} |
  {{Description |} <description>} |
  {{Exclude | Exc} <folders>} |
  {{Export | Exp} <VCM archive file>} |
  {{FixFloatingChildShares | True | False} |
  {{IgnoreMergePoints | IMP} [True | False]} |
  {{Include | Inc} {<change requests> | <files> | <folders> | <process items> |
    <requirements> | <revision labels> | <tasks> | <topics>}} |
  {{LockMergeConflicts | LMC} {None | Source | Target | Both}} |
  {{ManualMergeFiles | MMF} [True | False]} |
  {{Match [Folder] *{<folder path> to <folder path>}} |
  {{MergeType | Type | MT} {Compare | Rebase | Promote | Replicate}} |
  {(Name |Na} <Change Package name>} |
  {{PostCommitLabel | PostCL} <label>} |
  {{PostCommitRevLabel | PostRL} <label>} |
```

```
    {{PreCommitLabel | PreCL} <label>} |
    {{PreCommitRevLabel | PreRL} <label>} |
    {{PreventDuplicateFilenames | PDF} [True | False]} |
    {{Project | Pro} <project>} |
    {{ReportDiffs | RD} [True | False]} |
    {{ReportUpdates | RU} [True | False]} |
    {Save [<VCM session file>]} |
    {{SourceLabel | SrcLabel | SL} <label>} |
    {{SourceState | SrcState | SS} <state>} |
    {{SourceTime | SrcTime | ST} {<timestamp> | Now}} |
    {{SourceView | Source | SV} <view>} |
    {{TargetLabel | TgtLabel | TL} <label>} |
    {{TargetState | TgtState | TS} <state>} |
    {{TargetView | Target | TV} <view} |
```

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

その他の構文要素

以下の表にアルファベット順にその他の構文要素を示します。

その他の構文要素

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

```
<change requests>
  {CR | CRs | ChangeRequests} {ALL | *<CR #>}
```

```
<Change Package name>
 {A name consisting of one or more characters}
```

```
<check-out options>
  [+cwf] [+eol {on | off | cr | lf}] [+filter {CGIMOU}] [+o] [+ro]
  [+rp <work folder path>]
```

```
<condition name>
  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
```

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

```
<files>
   [File | Files] {ALL | *{<file name pattern>} [+<depth>]}
```

```
<folder path>
{A slash followed by an optional series of folder names each ending with a slash}
```

```
<folders>
  [Folder | Folders] {ALL | *{<folder path>} [+<depth>] *[<item type>]}
```

```
<item condition>
  <condition name> [<condition value>]
```

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

```
<match state>
  *<item condition>
```

```
<process items>
  ProcessItems *{[View <view>] CR <CR #> |
                 [View <view>] Req <Req #> |
                 [View <view>] Task <Task #>}
```

```
<requirements>
  {Requirement | Req | Requirements | Reqs} {ALL  | *<Req #>}
```

```
<revision labels>
  RevLabels *<label>
```

```
<task>
  {Task | Tasks} {ALL | *<Task #>}
```

```
<timestamp>
  Example formats:
    "3/11/06 1:32 PM"
    "Mar 11, 2006 1:32:38 PM"
    "March 11, 2006 1:32:38 PM PST"
    "Saturday, March 11, 2006 1:32:38 PM PST"
```

```
<topics>
  {Topic | Topics} {ALL | *<Topic #>}
```

```
<VCM exchange file>
A .vcms file name}
```

```
<VCM session file>
A .vcms file name}
```

関連参照

VCM コマンドライン ユーティリティの概要(VCMUtility)
VCMUtility のコマンド
VCMUtility の接続オプション
VCMUtility のセッション オプション
VCMUtility のその他のオプション
VCMUtility の使用例
VCMUtility の複合オプションの構文
<action>(アクション)
<change requests>(変更要求)
<check-out options>(チェックアウト オプション)
<files>(ファイル)
<folders>(フォルダ)
<process item>(処理アイテム)
<requirements>(要件)
<revision labels>(リビジョン ラベル)
<tasks>(タスク)
<timestamp>(タイムスタンプ)
<topics>(トピック)

VCMUtility の複合オプションの構文

VCMUtility オプションには、以下の複合オプションが使用できます。

このセクションの内容

<action>（アクション）

ソース アイテムとターゲット アイテムの差分に対して実行するアクションを指定します。<action>には以下のものがあります。

オプション	説明
Delete（削除）	ターゲット アイテムを削除します。
DeleteAndReverseShare（削除とリバース共有）	Delete 実行後、ReverseShare を実行します。
Fail（失敗）	NeedsReview と同意です（以下参照）。
Ignore（無視する）	アクションを実行しません。
MarkResolved（解決済みとしてマーク）	解決済みとマークされているソースとターゲットのアイテムを用いてマージ ポイントを作成します。
Merge（マージ）	ソースとターゲットのアイテムをマージします。
Move（移動）	ターゲットのアイテムをソース アイテム同様のフォルダに移動します。
MoveAndMerge（移動とマージ）	Move 実行後、Merge を実行します。
MoveAndOverwrite（移動と上書き）	移動 と後に 上書き を実行します。
MoveAndRepin（移動と再固定）	Move 実行後、Repin を実行します。
NeedsReview（要確認）	コミット前に確認を強制します。つまり、このアクションが選択されている間はコミットが許可されません。アイテムの差分に対して、（割り当てられた）アクションが競合しています。アクションを変更する必要があります。
Overwrite（上書き）	ソース アイテムの内容でターゲット アイテムを上書きします。
Repin（再固定）	ソース アイテムと内容を一致させるため、ターゲット アイテムをソース アイテムのリビジョンに変更します（固定する）。
ReverseShare（リバース共有）	ソース アイテムをターゲット ビューに移動し、ソースのビューに対して共有を返します。
Share（共有）	ソース アイテムをターゲット ビューに共有します。

注: あらゆる差分に対して、すべての アクション が有効であるとは限りません。たとえば、削除はターゲットのアイテムが既に削除されている場合、有効ではありません。

関連参照

VCMUtility の複合オプションの構文
<change requests>（変更要求）
<check-out options>（チェックアウト オプション）
<files>（ファイル）
<folders>（フォルダ）
<match state>（一致条件）
<process item>（処理アイテム）
<requirements>（要件）
<revision labels>（リビジョン ラベル）
<tasks>（タスク）
<timestamp>（タイムスタンプ）
<topics>（トピック）
クイック参照

<check-out options>(チェックアウト オプション)

このセクションでは、VCMUtility の <check-out options> を、パラメータを組み合わせて使用する際の構文について説明します。

[+cwf] [+eol {on | off | cr | lf | crlf}] [+filter {CGIMOU}] [+o] [+ro] [+rp <work folder path>]

デフォルト以外のチェックアウト オプションを指定します。使用できるチェックアウト オプションは、StarTeam のコマンドライン(stcmd:StarTeam Command-Line)および一括チェックアウト(BCO:Bulk Check-Out)ユーティリティと類似していますが、'+' をパラメータの前に付ける部分が異なります。利用可能なオプションの詳細を以下に示します。

+cwf

指定したフォルダに対して、作業フォルダを作成します。作業フォルダはチェックアウトするファイルがない場合でも作成されます。可視のフォルダのみが作成されます。

+eol <eol option>

テキストファイルの改行文字(EOL)を指定する形式に変換します。<eol option> を on にすると、クライアント環境の改行文字を使用します。off にすると、改行文字を変換しません。cr、lf、および crlf を用いると、改行文字はそれぞれ復帰、改行、または復帰/改行に変換されます。「固定」EOL 形式を持つテキスト ファイルは、常に特定の形式に変換される点に留意してください。

+filter

+filter {CGIMOU}

ファイルのステータスを指定し、該当するファイルだけをチェックアウトします。**C**urrent(最新)、mer**G**e(マージ)、m**I**ssing(作業ファイルなし)、**M**odified(変更済み)、**O**ut-of-date(古いリビジョン)、**U**nknown(不明)が選択できます。ステータスは組み合わせて使用できます。+filter を指定しない場合、デフォルトのフィルタは IO(**Missing(作業ファイルなし)** および **Out-of-date(古いリビジョン)**)となります。**Merge(マージ)**、**Merge(マージ)**、または **Unknown(不明)** が +o を使わずに指定された場合、該当のファイルに対して警告が生成され、ファイルはチェックアウトされません。

+o

Missing(不明) および **Out-of-date(古いリビジョン)** のファイルに加え、**Modified(変更済み)**、**Merge(マージ)**、または **Unknown(不明)** のファイル ステータスに対して指定します。すべてのファイルは警告なしに上書きされます。+filter を使用している場合、フィルタで指定したファイルのみをチェックアウトします。

+ro

チェックアウトしたファイルを読み取り専用にします。デフォルトではチェックアウトしたファイルは読み取り/書き込みが可能です。

+rp

"マージ プレビュー" のルート作業フォルダを指定します。ファイルは <work folder path> で指定する作業フォルダの子フォルダにチェックアウトされます。

関連参照

VCMUtility の複合オプションの構文
<action>(アクション)
<change requests>(変更要求)
<files>(ファイル)
<folders>(フォルダ)
<match state>(一致条件)
<process item>(処理アイテム)
<requirements>(要件)
<revision labels>(リビジョン ラベル)
<tasks>(タスク)
<timestamp>(タイムスタンプ)
<topics>(トピック)
クイック参照

<change requests>(変更要求)

<change requests> {CR | CRs | ChangeRequests} {ALL | *<CR #>}

ビュー内のすべての変更要求、または個別の変更要求を、変更要求番号で指定します。CRs および ChangeRequests どちらも使用できます。単数形(s をつけない)も使用可能です。

関連参照

VCMUtility の複合オプションの構文
<action>(アクション)
<check-out options>(チェックアウト オプション)
<files>(ファイル)
<folders>(フォルダ)
<match state>(一致条件)
<process item>(処理アイテム)
<requirements>(要件)
<revision labels>(リビジョン ラベル)
<tasks>(タスク)
<timestamp>(タイムスタンプ)
<topics>(トピック)
クイック参照

<files>（ファイル）

[File | Files] {ALL | *{<file name pattern> [+depth>]}}

ビュー内のすべてのファイル、または特定のファイル群を、ファイル名のリストおよび/またはパターンを用いて指定します。それぞれ、任意でフォルダの深さを <depth> で一緒に指定します。キーワード File（または Files）は、キーワード All が指定されない限り、任意指定です。<file name pattern> には、固有のファイル名（例、foo.java）、ファイル名のパターン（例、*.java）、または、フォルダ パス付きのファイル名やパターン（例、/src/com/acme/foo.java や /src/com/acme/*.java）を指定できます。

使用方法

フォルダのパスにはスラッシュ（/）を使用します。先頭の単一スラッシュ（/）はルート フォルダを表します（他の StarTeam ユーティリティとの一貫性を保つため、ルートフォルダ名（通常はビュー名と一致）には、パス名を**使用しません**）。

- ファイル名、パターンにフォルダ パスがない場合、直前に指定した <file name pattern> パラメータのフォルダを基準とします。
- 一番最初のオプションに <file name pattern> パラメータが指定されない場合、ルートフォルダが基準となります。
- どの深さレベルまで子フォルダを含めるかを <depth> で指定します。<depth> には数値または All を用いることができます。
- ファイル名、パターン名にスペースが含まれる場合、引用符で囲む必要があります。

使用例

以下に <files> の使用例を示します。

```
// all files in the view
include Files ALL

//foo.java and bar.java in folder /src/com/acme
include /src/com/acme/foo.java bar.java

// all .java files in folder /src/com/acme and below
include /src/com/acme/*.java +all

// all .txt files in the root folder, all .zip file in first-level
// child folders, and a specific readme.txt file
include *.txt *.zip +1 /docs/acme/readme.txt
```

関連参照

VCMUtility の複合オプションの構文
<action>(アクション)
<change requests>(変更要求)
<check-out options>(チェックアウト オプション)
<folders>(フォルダ)
<match state>(一致条件)
<process item>(処理アイテム)
<requirements>(要件)
<revision labels>(リビジョン ラベル)
<tasks>(タスク)
<timestamp>(タイムスタンプ)
<topics>(トピック)
クイック参照

<folders>（フォルダ）

[Folder | Folders] {ALL | *{<folder path> [+<depth>] *[<item type>]}}

ビュー、もしくは特定のフォルダ パスにあるすべてのフォルダを指定します。パラメータにより、フォルダの深さやアイテムのタイプを指定することもできます。キーワード Folder（または Folders）は、キーワード All が指定されない限り、任意指定です。

使用方法

<folder path> を指定する際は、最初と最後にスラッシュ（/）を付ける必要があります（例：/src/com/）。どの深さレベルまで子フォルダを含めるかをフォルダの <depth> で指定します。<depth> には数値または All を用いることができます。

- フォルダ パスにスペースが含まれる場合、引用符で囲む必要があります。
- <item type> パラメータを使用しない場合、指定フォルダにあるファイルのみが含まれます。それ以外の場合、指定アイテムのみが含まれます。

認識可能なアイテムのタイプは CRs（変更要求）、Files（ファイル）、Folders（フォルダ）、Tasks（タスク）、Topics（トピック）、Requirements（要件）です（単数形でも複数形でも構いません）。

使用例

以下に <folder> の使用例を示します。

```
// all folders in the view
include folders ALL

// all files in the folder /src/com/acme/ alone
include /src/com/acme/

// all files and tasks in  /src/ and below
include /src/ +all files tasks

// all CRs in the folder "/triage/" and all files in "/PR docs/"
// child folders two levels below it
include /triage/ CRs "/PR docs/" +2
```

慣例により、ルート フォルダは単一の "/" で表します。つまり、ルート フォルダ名はフォルダ パスで指定しないということです。たとえば、ルート フォルダ名が "StarDraw" である場合、直下の子フォルダ "Source Code" に対するフォルダ パスは単に /Source Code/ です。

関連参照

VCMUtility の複合オプションの構文
<action>(アクション)
<change requests>(変更要求)
<check-out options>(チェックアウト オプション)
<files>(ファイル)
<match state>(一致条件)
<process item>(処理アイテム)
<requirements>(要件)
<revision labels>(リビジョン ラベル)
<tasks>(タスク)
<timestamp>(タイムスタンプ)
<topics>(トピック)
クイック参照

<item type>(アイテム タイプ)

<item type>

アイテムのタイプを指定します。使用可能は値は、ChangeRequest(または CR)、File、Folder、Requirement(または Req)、Task、Topic です。アイテムのタイプ名では、大文字/小文字が区別され、複数形を指定することも可能です。

関連参照

VCMUtility の複合オプションの構文
<action>(アクション)
<change requests>(変更要求)
<check-out options>(チェックアウト オプション)
<files>(ファイル)
<folders>(フォルダ)
<process item>(処理アイテム)
<requirements>(要件)
<revision labels>(リビジョン ラベル)
<tasks>(タスク)
<timestamp>(タイムスタンプ)
<topics>(トピック)
クイック参照

<match state>(一致条件)

*<item condition>

ソース アイテム/ターゲット アイテムの差分に対して、適用するコンディションを定義します。<match state> は、それぞれの <item condition> の定義をまとめたものです。それぞれの <item condition> には書式が定められています。

<condition name> [<condition value>]

<condition name>(コンディション名)

使用可能な <condition names> とその意味は以下のとおりです。

<condition name>	意味
items.binaryfile	対象のアイテムがバイナリ ファイルであることを示します。
items.branched	ソースのアイテムとターゲットのアイテムがバージョン ツリーの異なる分岐にあることを示します。
items.samecontent	ソースのアイテムとターゲットのアイテムが変更可能な同じプロパティを持っていることを示します。対象がファイルの場合は、データ内容が一致していることを示します。
source.childshare	ソースのアイテムがターゲット アイテムの共有の子であることを示します。
source.deleted	対象のアイテムがソース ビューから削除されていることを示します。
source.floating	ソースのアイテムが変動構成であることを示します。
source.modified	対象アイテムがソース ビューで変更されていることを示します。
source.moved	対象のアイテムがソース ビューで移動していることを示します。
source.present	対象のアイテムがソース ビューに存在していることを示します。
source.rootbranch	ソースのアイテムが共有ツリーのルート分岐であることを示します。
target.childshare	ターゲットのアイテムがソース アイテムの共有の子であることを示します。
target.deleted	対象のアイテムがターゲット ビューから削除されていることを示します。
target.floating	ターゲットのアイテムが変動構成であることを示します。
target.modified	対象のアイテムがターゲット ビューで変更されていることを示します。
target.moved	対象のアイテムがターゲット ビューで移動していることを示します。
target.parentdeleted	ターゲットのアイテムが存在したフォルダが削除されていることを示します。
target.present	対象のアイテムがターゲット ビューに存在することを示します。
target.rootbranch	ターゲットのアイテムが共有ツリーのルート分岐であることを示します。

<condition value>(コンディションの値)

使用可能な <condition value> は以下のとおりです。

<condition value>	意味
True	アイテムのコンディションを true(一致)と指定します。
False	アイテムのコンディションを false(不一致)と指定します。
Unspecified	アイテムのコンディションが分からない場合や値を指定する必要がないことを示します。

<condition value> は任意指定のパラメータで、デフォルトの値は True です。<match state> においてコンディションを指定しないものについては、値に Unspecified が使われます。

<item condition> を True または False にすることで、条件が一致した実際のアイテムの差分を(特定のアクションを実行するなどの)対象にすることができます。

コンディションに Unspecified を指定すると、たとえば、オプション ファイルからコンディションを削除することなく、実験的に一致条件からコンディションを除くことができます。

注: 同時に指定できないコンディションも存在します。両方を指定すると、アイテムの差分にコンディションが合致することがなくなります。例として、ソース アイテムが存在し(source.present=true)、かつ、削除されている(source.deleted=true)ケースが挙げられます。

関連参照

VCMUtility の複合オプションの構文
<action>(アクション)
<change requests>(変更要求)
<check-out options>(チェックアウト オプション)
<files>(ファイル)
<folders>(フォルダ)
<process item>(処理アイテム)
<requirements>(要件)
<revision labels>(リビジョン ラベル)
<tasks>(タスク)
<timestamp>(タイムスタンプ)
<topics>(トピック)
クイック参照

<process item>(処理アイテム)

ProcessItems *{[View <view>] CR <CR #> | [View <view>] Req <Req #> | [View <view>] Task <Task #>}

スコープに含める処理アイテム(変更要求、タスク、要件)を指定します(複数の処理アイテムの指定も可能)。ソース ビューには指定した処理アイテムのほか、リンク アイテムが含まれます。キーワードの ProcessItems には単数形(s をつけない)も使えます。ChangeRequest (変更要求) および Requirement (要件) には、それぞれの省略形、CR および Req を使うことができます。

デフォルトの状態では、処理アイテムはソース ビューに存在している必要があります。しかし、オプションのパラメータ View <view> を用いることにより、ソース ビュー以外の処理アイテムを選択することができます。ソース ビュー以外の処理アイテムを含める場合、これらの処理アイテムがソース スコープに含まれることは**ありません**。しかし、ソース ビューに存在するアイテムが、リンクしている処理アイテムはスコープに含まれます。スコープに含まれるリビジョンは、ソース ビューに存在するアイテムのうち、処理アイテムにリンクされているものです。

例:

```
// Include CR #451 in the source view and its linked items
include ProcessItem CR 451

//Include the items in the source view that are linked to Task #909
//inclulde Requirement #518, both from view "Triage"
//include ProcessItem View Triage Task 909
                    View Triage Requirement 518
```

注: ビュー名にスペースが含まれる場合、引用符で囲む必要があります("Release 4.3")。プロジェクトに同じビュー名が 2 つ以上使われている場合、ビュー名をスラッシュ(/)で区切ります("Apps/Releases/Release 4.3")。

関連参照

- VCMUtility の複合オプションの構文
- <action>(アクション)
- <change requests>(変更要求)
- <check-out options>(チェックアウト オプション)
- <files>(ファイル)
- <folders>(フォルダ)
- <match state>(一致条件)
- <requirements>(要件)
- <revision labels>(リビジョン ラベル)
- <tasks>(タスク)
- <timestamp>(タイムスタンプ)
- <topics>(トピック)
- クイック参照

<requirements>(要件)

{Reqs | Requirements} {ALL | *{<Req #>}

要件(Requirements)の番号を用いて個々の要件を指定します。Reqs および Requirements どちらも使用できます。単数形(s をつけない)も使用可能です。

関連参照

VCMUtility の複合オプションの構文
<action>(アクション)
<change requests>(変更要求)
<check-out options>(チェックアウト オプション)
<files>(ファイル)
<folders>(フォルダ)
<match state>(一致条件)
<process item>(処理アイテム)
<revision labels>(リビジョン ラベル)
<tasks>(タスク)
<timestamp>(タイムスタンプ)
<topics>(トピック)
クイック参照

<revision labels>(リビジョン ラベル)

RevLabels *<label>

リビジョン ラベル(<label>)を指定し、そのラベルに添付されたアイテムを指定します。キーワードの RevLabels には単数形(RevLabel)も使えます。リビジョン ラベルには大文字小文字の区別がありません。

関連参照

- VCMUtility の複合オプションの構文
- <action>(アクション)
- <change requests>(変更要求)
- <check-out options>(チェックアウト オプション)
- <files>(ファイル)
- <folders>(フォルダ)
- <match state>(一致条件)
- <process item>(処理アイテム)
- <requirements>(要件)
- <tasks>(タスク)
- <timestamp>(タイムスタンプ)
- <topics>(トピック)
- クイック参照

<tasks>(タスク)

Tasks {ALL | *{<Task #>}

タスク(Tasks)の番号を用いて個々のタスクを指定します。キーワードの Tasks には単数形(Task)も使えます。

関連参照

VCMUtility の複合オプションの構文
<action>(アクション)
<change requests>(変更要求)
<check-out options>(チェックアウト オプション)
<files>(ファイル)
<folders>(フォルダ)
<match state>(一致条件)
<process item>(処理アイテム)
<requirements>(要件)
<revision labels>(リビジョン ラベル)
<timestamp>(タイムスタンプ)
<topics>(トピック)
クイック参照

<timestamp>（タイムスタンプ）

<timestamp> には Java が認識できる日付、時間の文字列形式を使用する必要があります。

- 日付形式の解釈方法は、ローカルの日付形式（たとえば、米国の 3/11/06 は 2006 年 3 月 11 日 と解釈されます）となります。
- 秒は任意設定項目です（たとえば、1:32 と 1:32:00 は同じ時間を表します）。
- AM/PM は必須設定項目です。
- タイムゾーンは任意設定項目で、省略された場合はローカルのタイム ゾーンが使われます。
- 曜日は設定しても無視されます。

例:

```
"3/11/06 1:32 PM"
"Mar 11, 2006 1:32:38 PM"
"March 11, 2006 1:32:38 PM PST"
"Saturday, March 11, 2006 1:32:38 PM PST"
```

関連参照

VCMUtility の複合オプションの構文
<action>（アクション）
<change requests>（変更要求）
<check-out options>（チェックアウト オプション）
<files>（ファイル）
<folders>（フォルダ）
<match state>（一致条件）
<process item>（処理アイテム）
<requirements>（要件）
<revision labels>（リビジョン ラベル）
<tasks>（タスク）
<topics>（トピック）
クイック参照

<topics>(トピック)

Topics {ALL | *{<Topic #>}

トピック(Topics)の番号を用いて個々のトピックを指定します。キーワードの Topics には単数形(Topic)も使えます。

関連参照

VCMUtility の複合オプションの構文
<action>(アクション)
<change requests>(変更要求)
<check-out options>(チェックアウト オプション)
<files>(ファイル)
<folders>(フォルダ)
<match state>(一致条件)
<process item>(処理アイテム)
<requirements>(要件)
<revision labels>(リビジョン ラベル)
<tasks>(タスク)
<timestamp>(タイムスタンプ)
クイック参照

索引