﻿== CMD\_DESCRIPTION\_ACL ==

設定物件權限。

== CMD\_USAGE\_ACL ==

用法：

cm ^acl (--^user=<使用者\_名稱> | --^group=<群組\_名稱>)

(-^allowed|-^denied|-^overrideallowed|-^overridedenied=+|-<權限>[,...])[,...]

<物件規格>

--^user 使用者名稱。

--^group 群組名稱。

-^allowed 啟用指定的一或多項權限。使用

逗號以分隔權限。(使用 'cm ^showpermissions'

以顯示所有可用的權限。)

-^denied 拒絕指定的一或多項權限。使用

逗號以分隔權限。(使用 'cm ^showpermissions'

以顯示所有可用的權限。)

-^overrideallowed 覆寫允許的一或多項權限。使用

逗號以分隔權限。(使用 'cm ^showpermissions'

以顯示所有可用的權限。)

-^overridedenied 覆寫拒絕的一或多項權限。使用

逗號以分隔權限。(使用 'cm ^showpermissions'

以顯示所有可用的權限。)

物件規格 將設定權限的物件。

此命令的有效物件為：

儲存庫伺服器、儲存庫、分支、變更集、標籤、項目，

以及屬性。

(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

安全路徑的特殊用法：

cm ^acl [(--^user=<使用者\_名稱> | --^group=<群組\_名稱>)

(-^allowed|-^denied|-^overrideallowed|-^overridedenied=+|-<權限>[,...])[,...]]

[--^delete] [--^branches=[+ | -]<分支>[,...]]

<規格>

--^delete 移除安全路徑。

如需詳細資訊，請參閱備註。

--^branches 將安全路徑權限設為分支群組。

使用逗號以分隔分支。

或者，每個分支前面可加上 + 或 -

簽署以指定在編輯時是否必須在清單中新增或刪除

分支。

如需詳細資訊，請參閱備註。

規格 設定權限所在的安全路徑。

== CMD\_HELP\_ACL ==

設定權限需要瞭解 Plastic SCM 安全性的運作方式。

查看《安全性指南》以瞭解權限的運作方式：

https://www.plasticscm.com/download/help/securityguide

備註：

此命令為使用者或群組設定特定物件、儲存庫、分支、標籤

及/或伺服器路徑的權限。

物件規格：

(使用 'cm ^help ^objectspec' 以瞭解如何指定物件。)

'^acl' 命令使用特殊類型的規格：安全路徑。

- 安全路徑規格：

^path:server\_path[#tag]

範例：^path:/src/foo.c

^path:/doc/pdf

^path:/doc/pdf#documents

權限動作：

使用 -^allowed 和 -^denied 以指定要設定的權限。

使用 -^overrideallowed 和 -^overridedenied 引數以指定

要覆寫的權限。

每個動作皆需要以逗號分隔的權限清單。

權限名稱：

每個權限前面已加上 + 或 - 符號。

+ 符號可設定權限，- 符號可清除權限。

若要查看物件的權限，請使用 'cm ^showacl' 命令。

已覆寫的權限：

使用 -^overrideallowed 和 -^overridedenied

覆寫權限可讓您略過繼承。

略過在儲存庫或伺服器層級設定的權限會

很有幫助。

範例：

cm ^acl --^user=vio -^allowed=+^ci -^overrideallowed=+^ci ^br:qa@test

(即使使用者 'vio' 的權限在儲存庫層級遭受拒絕，該使用者也可以在儲存庫 'test'

的分支 'qa' 上簽入。)

伺服器路徑權限 (又稱為安全路徑)：

- 可針對特定伺服器路徑指定權限。

- 簽入作業期間會檢查這些權限。

- 更新作業期間也會檢查這些權限，

並且可用來防止某些目錄和檔案

下載至工作區。

- 對於要簽入的每個項目，伺服器會嘗試將項目路徑

與安全路徑進行比對。如果相符，簽入作業會檢查

該項目是否具有要簽入的權限。

可以針對安全路徑定義的權限

如下：

'^ci'、'^change'、'^add'、'^move'、'^rm'、'^read'

如果任何涉及項目的權限檢查不成功，

將會復原簽入作業。

若要將安全路徑權限設為分支群組，請使用

--^branches 選項。

範例：

cm ^acl --^user=jo -^denied=+^ci ^path:/src#rule0 --^branches=main,main/rel0

若要編輯與安全路徑相關的 ACL，標籤很有用。

範例：

cm ^acl --^user=jo -^denied=+^rm ^path:/src#rule0

(如果沒有此標籤，可能需要再次指定

分支清單。)

可編輯安全路徑的分支清單。

範例：

cm ^acl ^path:/src#rule0 --^branches=-main,+main/rel1

(將 'main' 從清單中移除，並新增 'main/rel1'。)

若要移除安全路徑，請使用 --^delete 引數。

範例：

cm ^acl --^user=jo --^delete ^path:/src#rule0

繼承：

繼承是來自 Plastic SCM 3.0 的選項。

這是進階選項，但幾乎已棄用。

該選項可讓物件繼承來自任何其他物件的權限，

覆寫預設的繼承關係。

使用選項 -^cut 以切斷繼承鏈。

使用選項 -^cutncpy 以切斷和複製目前繼承的

權限。(這是受到 Windows 檔案系統權限的啟發

，在當中您可以切斷繼承，但保留實際權限。)

-^inherit 選項可讓使用者繼承物件規格。

範例：'-^inherit=物件\_規格'

範例：

cm ^acl --^user=danipen -^denied=+^ci ^rep:core

(在儲存庫 'core' 上拒絕使用者 'danipen' 的簽入。)

cm ^acl --^group=developers -^allowed=+^view,-^read -^denied=+^chgperm ^br:main

(此命令可針對 'main' 分支中 'developers' 群組授予檢視權限、

清除讀取權限，並拒絕 chgperm 權限。)

安全路徑範例：

cm ^acl --^group=devs -^denied=+^ci ^path:/server#rel --^branches=main,main/2.0

(此命令會拒絕分支 'main' 和 'main/2.0' 中任何符合 '/server' 之路徑

的 '開發' 群組的簽入權限。建立標籤 '#rel'

是為了能在日後進行參照。)

cm ^acl ^path:/server#rel --^branches=-/main,+/main/Rel2.1

(更新安全路徑 '/server'，其標籤是 'rel'，移除

'main' 分支，並將分支 'main/Rel2.1' 新增至

安全路徑適用的分支群組。請考量先前範例，

現在分支清單將包含 'main/Rel2.1' 和 'main/2.0'。)

cm ^acl --^user=vsanchezm -^allowed=-^read -^overrideallowed=+^read ^path:/doc

(針對 '/doc' 路徑中覆寫的 'vsanchezm'，移除 '^read' 權限。)

== CMD\_DESCRIPTION\_ACTIVATEUSER ==

啟動授權使用者。

== CMD\_USAGE\_ACTIVATEUSER ==

用法：

cm ^activateuser | ^au <使用者名稱>[ ...][--^server=<儲存庫伺服器規格>]

使用者名稱 要啟動的一或多個使用者名稱。使用雙引號 (" ")

以指定使用者名稱 (包含空格)。使用空格

以分隔使用者名稱。

選項：

--^server=<儲存庫伺服器規格> 在指定的伺服器中啟動使用者。

如果未指定任何伺服器，請在 client.conf 檔案的

預設伺服器中執行命令。

(使用 'cm ^help ^objectspec' 以深入瞭解

儲存庫伺服器規格。)

== CMD\_HELP\_ACTIVATEUSER ==

備註：

若要啟動使用者，該使用者先前必須已停用。

預設情況下，第一次在 Plastic SCM 中執行寫入

作業時，會啟動使用者。只有在未超過使用者人數上限時，

才會自動啟動使用者。

如需停用 Plastic SCM 使用者的詳細資訊，請參閱

'cm ^help ^deactivateuser' 命令。

範例：

cm ^activateuser john

cm ^activateuser david "mary collins"

cm ^au peter --^server=localhost:8087

== CMD\_DESCRIPTION\_ADD ==

將項目新增至版本控制。

== CMD\_USAGE\_ADD ==

用法：

cm ^add [-^R | -^r | --^recursive] [--^silent] [--^ignorefailed]

[--^skipcontentcheck] [--^coparent] [--^filetypes=<檔案>] [--^noinfo]

[--^format=<字串格式>] [--^errorformat=<字串格式>]

<項目路徑>[ ...]

項目路徑 要新增的一或多個項目。使用雙引號 (" ") 指定

含空格的路徑。使用空格以分隔項目。

使用 \* 以新增目前目錄的所有內容。

選項：

-^R -^r --^recursive 遞迴新增項目。

--^silent 不顯示任何輸出。

--^ignorefailed 如果無法新增項目，新增作業將會在沒有項目的情況下，

繼續進行新增作業。請注意：如果無法新增目錄

，則不會新增其內容。

--^skipcontentcheck 當擴充功能不足以將檔案設為文字

或二進位時，將設為二進位，而不會

檢查內容以偵測類型。這麼做的目的是

提高大量簽入的效能。

--^coparent 針對要新增項目的上層執行簽出。

--^filetypes 要使用的檔案類型檔案。如需詳細資訊，

請查看以下連結：

http://blog.plasticscm.com/2008/03/custom-file-types.html

--^noinfo 不列印進度資訊。

--^format 以指定的格式擷取輸出訊息。如需詳細資訊，

請查看範例。

--^errorformat 以指定的格式擷取錯誤訊息

(如果有的話)。如需詳細資訊，請查看範例。

== CMD\_HELP\_ADD ==

備註：

新增項目的要求：

- 要新增之項目的上層目錄先前必須已新增。

從 stdin 讀取輸入：

'^add' 命令可從 stdin 讀取路徑。若要執行此操作，可傳遞單一破折號

「-」來完成。

範例：cm ^add -

將讀取路徑，直到輸入空白行為止。

這可讓您使用管線以指定要新增的檔案。

範例：

dir /S /B \*.c | cm ^add -

(在 Windows 中，新增工作區中所有 .c 檔案。)

範例：

cm ^add file1.txt file2.txt

(新增 'file1.txt' 和 'file2.txt' 項目。)

cm ^add c:\workspace\file.txt

(在路徑 'c:\workspace' 中新增 'file.txt' 項目。)

cm ^add -^R c:\workspace\src

(遞迴新增 'src'。)

cm ^add -^R \*

(遞迴新增目前目錄的所有內容。)

cm ^add -^R \* --^filetypes=filetypes.conf

(遞迴新增目前目錄的所有內容，使用

'filetypes.conf' 以根據其副檔名將類型指派給每個檔案，

而不是檢查其內容。)

cm ^add --^coparent c:\workspace\dir\file.txt

(將 'file.txt' 新增至原始檔控制，並執行 'dir' 的簽出。)

cm ^add -^R \* --^format="ADD {0}" --^errorformat="ERR {0}"

(遞迴新增目前目錄的所有內容，對於成功新增的檔案，

列印 '^ADD <項目>'，對於無法新增的項目，列印

'^ERR <項目>'。)

== CMD\_USAGE\_ADDIGNOREPATTERN ==

用法：

cm ^addignorepattern <模式>[ ...]

[--^workspace=<工作區路徑> | --^allworkspaces] [--^remove]

== CMD\_DESCRIPTION\_ADMIN ==

在伺服器上執行系統管理命令。

== CMD\_USAGE\_ADMIN ==

用法：

cm ^admin <命令> [選項]

可用命令：

^readonly

若要取得每個命令的詳細資訊，請執行：

cm ^admin <命令> --^usage

cm ^admin <命令> --^help

== CMD\_HELP\_ADMIN ==

備註：

只有伺服器管理員才可以執行系統管理命令。

範例：

cm ^admin ^readonly ^enter

cm ^admin ^readonly ^status

== CMD\_DESCRIPTION\_ADMIN\_READONLY ==

啟用/停用伺服器唯讀模式。

== CMD\_USAGE\_ADMIN\_READONLY ==

用法：

cm ^admin ^readonly (^enter | ^leave | ^status) [<伺服器>]

動作：

^enter 伺服器進入唯讀模式。寫入作業將遭受拒絕。

^leave 伺服器離開唯讀模式。

^status 顯示伺服器唯讀模式狀態。

選項：

伺服器 在指定的伺服器中執行命令 (server:port)。(使用

'cm ^help ^objectspec' 以深入瞭解伺服器規格。)

如果未指定任何伺服器，命令將使用目前

工作區中的伺服器。

如果目前路徑不在工作區中，命令會使用 client.conf

設定檔案中定義的預設伺服器。

== CMD\_HELP\_ADMIN\_READONLY ==

備註：

只有伺服器管理員才可以進入伺服器唯讀模式。

範例：

cm ^admin ^readonly ^enter diana:8086

cm ^admin ^readonly ^leave

== CMD\_DESCRIPTION\_ANNOTATE ==

顯示檔案每一行上次修改的變更集及其作者。

== CMD\_USAGE\_ANNOTATE ==

用法：

cm ^annotate | ^blame <規格>[ ...]

[--^format=<字串\_格式>]

[--^ignore=(^eol | ^whitespaces | ^"eol&whitespaces" | ^none)]

[--^dateformat=<字串\_日期\_格式>]

[--^encoding=<名稱>]

[--^stats]

[--^repository=<儲存庫規格>]

規格 要標註的檔案規格。

(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

使用雙引號 (" ") 指定含空格的路徑。

選項：

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

--^ignore 設定指定的比較方法。

如需詳細資訊，請參閱備註。

--^dateformat 設定輸出格式以列印日期。

--^encoding 指定輸出編碼，例如：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

--^stats 顯示統計資料資訊。

--^repository 指定用於計算

標註的儲存庫規格。預設情況下，此命令會使用

儲存庫，其中，工作區中載入的修訂儲存庫

即存放在此。(使用 'cm ^help ^objectspec' 以深入

瞭解儲存庫規格。)

== CMD\_HELP\_ANNOTATE ==

備註：

二進位檔案無法標註。

--^ignore 選項：

^none 偵測行尾和空格差異。

^eol 忽略行尾差異。

^whitespaces 忽略空格差異。

^"eol&whitespaces" 忽略行尾和空格差異。

--^format 選項：

此命令的輸出參數如下所示：

{^owner} 上次變更行的使用者。

{^rev} 行的來源修訂規格。

{^content} 行內容。

{^date} 簽入行的日期。

{^comment} 行的來源修訂註解。

{^changeset} 行的來源修訂變更集。

{^line} 檔案的行號。

{^id} 項目 ID。

{^parentid} 項目的上層 ID。

{^rep} 項目的儲存庫。

{^branch} 行的來源修訂分支。

{^ismergerev} 此行的修訂是否在合併中建立。

--^dateformat:

若要指定列印日期的輸出格式。

請參閱指定的支援格式，網址為：

https://docs.microsoft.com/en-us/dotnet/standard/base-types/custom-date-and-time-format-strings

--^repository:

可從遠端儲存庫擷取資料。對於分散式

案例很有用。

範例：

cm ^blame c:\workspace\src --^ignore=^"eol&whitespaces" --^encoding=utf-8

cm ^annotate c:\workspace\file.txt --^ignore=^eol

cm ^annotate c:\workspace\file.txt --^format="{^owner} {^date, 10} {^content}"

(依序寫入擁有者欄位、空白、日期欄位 (靠右對齊)，

然後空白，再寫入內容。)

cm ^blame c:\workspace\file.txt --^format="{^owner, -7} {^comment} {^date}" \

--^dateformat=yyyyMMdd

(以 7 個空格寫入擁有者欄位 (靠左對齊)，接著是

空白、註解，然後是另一個空白，最後是

已格式化的日期 (例如，20170329)。)

cm ^annotate c:\workspace\file.txt --^repository=centralRep@myserver:8084

cm ^blame ^serverpath:/src/client/checkin/Checkin.cs#^cs:73666

(使用伺服器路徑針對以變更集 73666 開始的檔案進行標註。)

== CMD\_DESCRIPTION\_APPLY\_LOCAL ==

檢查本地變更 (本地移動、本地刪除，以及本地變更)

並加以套用，這樣 Plastic SCM 就會開始追蹤這些變更。

== CMD\_USAGE\_APPLY\_LOCAL ==

用法：

cm ^applylocal | ^al [--^dependencies] [<項目\_路徑>[ ...]]

[--^machinereadable [--^startlineseparator=<分隔符號>]

[--^endlineseparator=<分隔符號>] [--^fieldseparator=<分隔符號>]]

選項：

--^dependencies 將本地變更相依性新增至

要套用的項目。

項目\_路徑 要套用的項目。使用空格分隔

路徑。使用雙引號 (" ") 指定

含空格的路徑。

--^machinereadable 以容易解析的格式輸出結果。

--^startlineseparator 搭配 '--^machinereadable' 標誌使用，指定

各行應如何起始。

--^endlineseparator 搭配 '--^machinereadable' 標誌使用，指定

各行應如何結束。

--^fieldseparator 搭配 '--^machinereadable' 標誌使用，指定應如何

分隔欄位。

== CMD\_HELP\_APPLY\_LOCAL ==

備註：

如果未指定 --^dependencies 和 <項目\_路徑>，該作業會涉及

工作區中所有本地變更。

一律會從指定路徑進行遞迴套用。

範例：

cm ^applylocal foo.c bar.c

cm ^applylocal .

(在目前目錄中套用所有本地變更。)

cm ^applylocal

(在工作區中套用所有本地變更。)

cm ^applylocal --^machinereadable

(在工作區中套用所有本地變更，並以

簡化的易於剖析的格式列印結果。)

cm ^applylocal --^machinereadable --^startlineseparator=">" \

--^endlineseparator="<" --^fieldseparator=","

(在工作區中套用所有本地變更，並以簡化、

易於剖析的格式列印結果，使用指定的字串來

起始和結束行，以及分隔欄位。)

== CMD\_DESCRIPTION\_ARCHIVE ==

外部儲存區中的封存資料。

== CMD\_USAGE\_ARCHIVE ==

用法：

cm ^archive | ^arch <修訂規格>[ ...][-^c | --^comment=<字串\_註解>]

[--^file=<基底\_檔案>]

(從儲存庫擷取資料，並將其存放在外部儲存區。)

cm ^archive | ^arch <修訂規格>[ ...]--^restore

(將先前封存的修訂還原至儲存庫。)

修訂規格 一或多個修訂規格。可以使用「-」修飾詞

從 STDIN 讀取。(使用 'cm ^help ^objectspec' 以

深入瞭解修訂規格。)

--^restore 從產生的封存檔案還原先前封存

的資料。

選項：

-^c | --^comment 在要建立的封存儲存檔案中設定註解。

--^file 新封存資料檔案的名稱首碼和 (選用)

路徑。

== CMD\_HELP\_ARCHIVE ==

備註：

此命令會從儲存庫資料庫擷取資料，並將其存放在

外部儲存區，節省資料庫空間。

此命令也會將先前封存的修訂還原至

儲存庫資料庫 (--^restore)。

使用 'cm ^help ^objectspec' 以瞭解如何指定修訂規格。

執行此命令的使用者必須是 Plastic SCM 伺服器管理員

(儲存庫伺服器擁有者)，才能完成此作業。

來自指定修訂的每個資料區段將存放在

不同檔案中，其名稱以 --^file

引數定義的值為開頭。該引數可以包含完整的路徑值，包括未來封存檔案

的首碼，也可以僅包括此首碼值。

封存後，指定修訂的資料可以透過兩種方式

存取：

- 從用戶端：用戶端將偵測資料是否已封存，而且

將提示使用者輸入檔案位置。

使用者可以藉由建立名為 externaldata.conf 的檔案

來設定外部資料位置 (在標準設定檔案位置，使用

適用於 client.conf 檔案的相同規則)，包含

封存資料所在的路徑。

- 從伺服器：使用此方法的使用者不必知道資料是否

封存，因為伺服器將會明確地解析

要求。若要執行此作業，管理員將會在伺服器目錄中

建立名為 externaldata.conf 的檔案，並填入

封存磁碟機所在的路徑。

若要取消封存 (還原) 修訂 (或修訂集合)，用戶端必須要能夠

存取封存的檔案。因此，無法取消封存

伺服器 (方法 2) 所解析的資料，因為用戶端

將無法將該資料識別為已封存。如果使用方法 2，若要

成功取消封存，管理員必須先編輯 externaldata.conf

伺服器檔案，才能移除必須取消封存之

封存檔案的存取權。

設定 PLASTICEDITOR 環境變數，以指定要輸入註解

的編輯器。

範例：

cm ^archive bigfile.zip#^br:/main

(在分支 'main' 中封存 'bigfile.zip' 的最後一個修訂。)

cm ^archive ^rev:myfile.pdf#^cs:2 -^c="大型 pdf 檔案" --^file=c:\arch\_files\arch

(封存 'c:\archived\_files' 資料夾中對 myfile.pdf 的變更集 2

的修訂。封存檔案名稱將以 'arch' 為開頭 (例如，arch\_11\_56)。)

cm ^find "^revs ^where ^size > 26214400" --^format="{^item}#{^branch}" \

--^nototal | cm ^archive --^comment="volume00" --^file="volume00" -

(將所有大於 25Mb 的所有檔案封存在名稱以 'volume00' 為開頭

的檔案中。)

cm ^find "^revs ^where ^size > 26214400 ^and ^archived='true'" \

--^format="{^item}#{^branch}" --^nototal | cm ^archive --^restore

(還原大於 25Mb 的所有封存檔案。)

== CMD\_DESCRIPTION\_ATTRIBUTE ==

可讓使用者管理屬性。

== CMD\_USAGE\_ATTRIBUTE ==

用法：

cm ^attribute | ^att <命令> [選項]

命令：

^create | ^mk

^delete | ^rm

^set

^unset

^rename

^edit

若要取得每個命令的詳細資訊，請執行：

cm ^attribute <命令> --^usage

cm ^attribute <命令> --^help

== CMD\_HELP\_ATTRIBUTE ==

範例：

cm ^attribute ^create 狀態

cm ^attribute ^set ^att:status ^br:/main/SCM105 未解決

cm ^attribute ^unset ^att:status ^br:/main/SCM105

cm ^attribute ^delete ^att:status

cm ^attribute ^rename ^att:status "組建狀態"

cm ^attribute ^edit ^att:status "CI 管線中的任務狀態"

== CMD\_DESCRIPTION\_CHANGELIST ==

分組變更清單中暫止的變更。

== CMD\_USAGE\_CHANGELIST ==

用法：

a) 變更清單物件的管理：

cm ^changelist | ^clist [--^symlink]

(顯示所有變更清單。)

cm ^changelist | ^clist ^add <變更清單\_名稱>

[<變更清單\_描述>] [--^persistent | --^notpersistent] [--^symlink]

(建立變更清單。)

cm ^changelist | ^clist ^rm <變更清單\_名稱> [--^symlink]

(移除選取的變更清單。如果此變更清單包含暫止的變更，

則這些將移動至 ^default 變更清單。)

cm ^changelist | ^clist ^edit <變更清單\_名稱> [<動作\_名稱> <動作\_值>]

[--^persistent | --^notpersistent] [--^symlink]

(編輯選取的變更清單。)

b) 管理指定變更清單的內容：

cm ^changelist | ^clist <變更清單\_名稱> (^add | ^rm) <路徑\_名稱>[ ...]

[--^symlink]

(藉由新增 ('^add') 或移除 ('^rm')

符合指定路徑名稱的變更，新增選取的變更清單。使用空格

分隔路徑\_名稱。使用雙引號 (" ") 指定含空格的

路徑。路徑狀態必須是 '^Added' 或 '^Checked-out'。)

選項：

變更清單\_名稱 變更清單的名稱。

變更清單\_描述 變更清單的描述。

動作\_名稱 選擇 '^rename' 或 '^description' 以編輯

變更清單。

動作\_值 編輯變更清單時，套用新名稱或

新描述。

--^persistent 即使簽入或復原變更清單的內容，變更清單

仍將保留工作區中。

--^notpersistent (預設) 即使簽入或復原變更清單的內容，

變更清單也不會保留在

工作區中。

--^symlink 將作業套用至符號連結，而不是套用至

目標。

== CMD\_HELP\_CHANGELIST ==

備註：

'^changelist' 命令會處理工作區暫止的變更清單以及

變更清單中包含的變更。

範例：

cm ^changelist

(顯示目前工作區變更清單。)

cm ^changelist ^add 組態\_變更 "dotConf 檔案" --^persistent

(建立名稱為 '組態\_變更' 且描述為 'dotConf 檔案'

的新變更清單，一旦暫止的變更清單已簽入或已復原，

該變更清單都將持續保留在目前工作區中。)

cm ^changelist ^edit 組態\_變更 ^rename 組態\_檔案 --^notpersistent

(編輯名稱為 '組態\_變更' 的變更清單，並將其重新命名為

'組態\_檔案'。此外，它會將變更清單轉變成「非持續」。)

cm ^changelist ^edit 組態\_變更 --^notpersistent

(編輯名稱為 '組態\_變更' 的變更清單，並將它轉變成「非持續」。)

cm ^changelist ^rm 組態\_檔案

(將暫止的變更清單 '組態\_檔案' 從目前工作區中移除。)

cm ^changelist 組態\_檔案 ^add foo.conf

(將檔案 'foo.conf' 新增至 '組態\_檔案' 變更清單。)

cm ^changelist 組態\_檔案 ^rm foo.conf readme.txt

(將檔案 'foo.conf' 和 'readme.txt' 從 '組態\_檔案' 變更清單中移除，

並將檔案移動至系統預設變更清單。)

== CMD\_DESCRIPTION\_CHANGESET ==

在變更集執行進階操作。

== CMD\_USAGE\_CHANGESET ==

用法：

cm ^changeset <命令> [選項]

命令：

^move | ^mv

^delete | ^rm

^editcomment | ^edit

若要取得每個命令的詳細資訊，請執行：

cm ^changeset <命令> --^usage

cm ^changeset <命令> --^help

== CMD\_HELP\_CHANGESET ==

範例：

cm ^changeset ^move ^cs:15@myrepo ^br:/main/scm005@myrepo

cm ^changeset ^delete ^cs:2b55f8aa-0b29-410f-b99c-60e573a309ca@devData

== CMD\_DESCRIPTION\_CHANGESET\_EDIT\_COMMENT ==

修改變更集的註解。

== CMD\_USAGE\_CHANGESET\_EDIT\_COMMENT ==

用法：

cm ^changeset ^editcomment | ^edit <變更集規格> <新\_註解>

選項：

變更集規格 將編輯註解的目標變更集。

(使用 'cm ^help ^objectspec' 以深入瞭解變更集

規格。)

新\_註解 將新增至鎖定目標之變更集的

新註解。

== CMD\_HELP\_CHANGESET\_EDIT\_COMMENT ==

備註：

- 鎖定目標的變更集規格必須有效。

範例：

cm ^changeset ^editcomment ^cs:15@myrepo "我忘記新增簽入詳細資料"

cm ^changeset ^edit ^cs:cb11ecdb-1aa9-4f11-8698-dcab14e5885a \

"此註解文字將取代先前的註解文字。"

== CMD\_DESCRIPTION\_CHANGESET\_MOVE ==

將變更集及其所有下層移動至不同的分支。

== CMD\_USAGE\_CHANGESET\_MOVE ==

用法：

cm ^changeset ^move | ^mv <變更集規格> <分支規格>

選項：

變更集規格 要移動至不同分支的第一個變更集。相同

分支中的所有下層變更集也會

由命令鎖定目標。

(使用 'cm ^help ^objectspec' 以深入瞭解變更集

規格。)

分支規格 存放鎖定目標之變更集的目標

分支。須為空白或不存在；如果

目的地分支不存在，將由命令

建立一個。

(使用 'cm ^help ^objectspec' 以深入瞭解分支

規格。)

== CMD\_HELP\_CHANGESET\_MOVE ==

備註：

- 鎖定目標的變更集規格必須有效。

- 目的地分支須為空白或不存在。

- 如果目的地分支不存在，將會建立一個。

- 合併連結將保持不變，因為分支不會影響這些連結。

範例：

cm ^changeset ^move ^cs:15@myrepo ^br:/main/scm005@myrepo

cm ^changeset ^move ^cs:cb11ecdb-1aa9-4f11-8698-dcab14e5885a ^br:/hotfix/TL-352

== CMD\_DESCRIPTION\_CHANGESET\_DELETE ==

從儲存庫中刪除變更集。

== CMD\_USAGE\_CHANGESET\_DELETE ==

用法：

cm ^changeset ^delete | ^rm <變更集規格>

選項：

變更集規格 要移除的目標變更集。必須滿足

某些特定條件。如需詳細資訊，請參閱備註。

(使用 'cm ^help ^objectspec' 以深入瞭解變更集

規格。)

== CMD\_HELP\_CHANGESET\_DELETE ==

備註：

- 目標變更集必須是其分支的最後一個。

- 目標變更集不可為任何其他變更集的上層。

- 目標變更集不能是合併連結的來源，也不能是

作為來源之間隔合併的一部分。

- 沒有標籤必須套用至目標變更集。

- 目標變更集不得為根變更集 ('^cs:0')

範例：

cm ^changeset ^rm ^cs:4525@myrepo@myserver

cm ^changeset ^delete ^cs:cb11ecdb-1aa9-4f11-8698-dcab14e5885a

== CMD\_DESCRIPTION\_CHANGEUSERPASSWORD ==

變更使用者密碼 (UP)。

== CMD\_USAGE\_CHANGEUSERPASSWORD ==

用法：

cm ^changepassword | ^passwd

== CMD\_HELP\_CHANGEUSERPASSWORD ==

備註：

此命令只有在安全性設定為 UP (使用者/密碼) 時

才可使用。如需詳細資訊，請參閱《管理指南》。

需要舊密碼與新密碼。

範例：

cm ^passwd

== CMD\_DESCRIPTION\_CHECKCONNECTION ==

檢查與伺服器的連線。

== CMD\_USAGE\_CHECKCONNECTION ==

用法：

cm ^checkconnection | ^cc

== CMD\_HELP\_CHECKCONNECTION ==

備註：

- 此命令會傳回訊息，指出設定的 Plastic SCM

伺服器是否具備有效連線。

- 此命令會檢查設定的使用者是否有效。它也會

檢查與伺服器的版本相容性。

== CMD\_DESCRIPTION\_CHECKDB ==

檢查儲存庫完整性。

== CMD\_USAGE\_CHECKDB ==

用法：

cm ^checkdatabase | ^chkdb [<儲存庫伺服器規格> | <儲存庫規格>]

使用 'cm ^help ^objectspec' 以深入瞭解儲存庫伺服器與儲存庫規格。

== CMD\_HELP\_CHECKDB ==

備註：

- 如果未指定儲存庫伺服器規格，也未指定儲存庫規格，

將會在 client.conf 檔案中指定的伺服器中執行檢查。

範例：

cm ^checkdatabase ^repserver:localhost:8084

cm ^chkdb ^rep:default@localhost:8084

== CMD\_DESCRIPTION\_CHECKIN ==

在此儲存庫中儲存變更。

== CMD\_USAGE\_CHECKIN ==

用法：

cm ^checkin | ^ci [<項目\_路徑>[ ...]]

[-^c=<字串\_註解> | -^commentsfile=<註解\_檔案>]

[--^all|-^a] [--^applychanged] [--^private] [--^update] [--^symlink]

[--^noshowchangeset]

[--^machinereadable [--^startlineseparator=<分隔符號>]

[--^endlineseparator=<分隔符號>] [--^fieldseparator=<分隔符號>]]

選項：

項目\_路徑 要簽入的項目。使用雙引號 (" ")

指定含空格的路徑。使用空格

分隔項目路徑。

使用 . 將簽入套用至目前目錄。

-^c 將指定的註解套用至

簽入作業中建立的變更集。

-^commentsfile 將指定檔案中的註解套用至

簽入作業中建立的變更集。

--^all | -^a 指定路徑上在本地變更、移動和刪除的項目

也包括在內。

--^applychanged 將簽入作業套用至

工作區中偵測到的變更項目，以及簽出

項目。

--^private 工作區中偵測到的私人項目也

包括在內。

--^update 如果最終發生此情況，

會自動處理更新合併。

--^symlink 將簽入作業套用至符號連結，而不是

套用至目標。

--^noshowchangeset 不列印結果變更集。

--^machinereadable 以容易解析的格式輸出結果。

--^startlineseparator 搭配 '--^machinereadable' 標誌使用，指定

各行應如何起始。

--^endlineseparator 搭配 '--^machinereadable' 標誌使用，指定

各行應如何結束。

--^fieldseparator 搭配 '--^machinereadable' 標誌使用，指定應

如何分隔欄位。

== CMD\_HELP\_CHECKIN ==

備註：

- 如果未指定 <項目\_路徑>，簽入會涉及工作區中

所有暫止的變更。

- 簽入作業一律會從指定路徑進行遞迴套用。

- 若要簽入項目：

- 項目必須受原始程式碼控制。

- 如果項目是私人的 (未受到原始程式碼控制)，則需要有 --^private

標誌，才能簽入該項目。

- 必須簽出項目。

- 如果項目已變更，但未簽出，則不一定要有 --^applychanged 標誌，

除非 <項目\_路徑> 是目錄，或包含

萬用字元 ('\*')。

修訂內容應該與先前修訂不同，才能

簽入。

設定 PLASTICEDITOR 環境變數，以指定要輸入

註解的編輯器。

從 stdin 讀取輸入：

'^checkin' 命令可從 stdin 讀取路徑。若要執行此操作，可傳遞單一

破折號「-」來完成。

範例：cm ^checkin -

將讀取路徑，直到輸入空白行為止。

這可讓您使用管線以指定要簽入的檔案。

範例：

dir /S /B \*.c | cm ^checkin --^all -

(在 Windows 中，簽入工作區中所有 .c 檔案。)

範例：

cm ^checkin file1.txt file2.txt

(簽入 'file1.txt' 和 'file2.txt' 簽出檔案。)

cm ^checkin .-^commentsfile=mycomment.txt

(簽入目前目錄，並在 'mycomment.txt' 檔案中

設定註解。)

cm ^checkin 連結 --^symlink

(簽入 '連結' 檔案，而非目標；適用於 UNIX

環境。)

cm ^ci file1.txt -^c="我的註解"

(簽入 'file1.txt' 並包含註解。)

cm ^status --^short --^compact --^changelist=暫止\_待\_檢閱 | cm ^checkin -

(列出變更清單中名稱為 '暫止\_待\_檢閱' 的路徑，並將此清單

重新導向簽入命令的輸入。)

cm ^ci .--^machinereadable

(簽入目前目錄，並以簡化、易於剖析

的格式來列印結果。)

cm ^ci .--^machinereadable --^startlineseparator=">" --^endlineseparator="<" --^fieldseparator=","

(簽入目前目錄，並以簡化、易於剖析的格式

來列印結果，使用指定的字串起始行

和結尾行，以及分隔欄位。)

== CMD\_DESCRIPTION\_CHECKOUT ==

將檔案標示為準備好修改。

== CMD\_USAGE\_CHECKOUT ==

用法：

cm ^checkout | ^co [<項目\_路徑>[ ...]] [-^R | -^r | --^recursive]

[--^format=<字串\_格式>]

[--^errorformat=<字串\_格式>] [--^resultformat=<字串\_格式>]

[--^silent] [--^symlink] [--^ignorefailed]

[--^machinereadable [--^startlineseparator=<分隔符號>]

[--^endlineseparator=<分隔符號>] [--^fieldseparator=<分隔符號>]]

選項：

項目\_路徑 要簽出的項目。使用雙引號 (" ")

指定含空格的路徑。使用空格

分隔項目路徑。

使用 . 將簽出套用至目前目錄。

-^R 遞迴簽出檔案。

--^format 以指定的

格式擷取輸出進度訊息。如需詳細資訊，請查看範例。

--^errorformat 以指定的

格式擷取錯誤訊息 (如果有的話)。如需詳細資訊，請查看範例。

--^resultformat 以指定的格式擷取輸出

結果訊息。如需詳細資訊，請查看範例。

--^silent 完全不顯示任何輸出。

--^symlink 將簽出作業套用至符號連結，而非

套用至目標。

--^ignorefailed 如果無法鎖定項目 (無法執行獨佔簽出

)，則會在沒有項目的情況下，

繼續進行簽出作業。

--^machinereadable 以容易解析的格式輸出結果。

--^startlineseparator 搭配 '--^machinereadable' 標誌使用，指定

各行應如何起始。

--^endlineseparator 搭配 '--^machinereadable' 標誌使用，指定

各行應如何結束。

--^fieldseparator 搭配 '--^machinereadable' 標誌使用，指定應

如何分隔欄位。

== CMD\_HELP\_CHECKOUT ==

備註：

若要簽出項目：

- 項目必須受原始程式碼控制。

- 必須簽入項目。

如果在伺服器上已設定鎖定 (lock.conf 存在)，那麼每次在路徑上

發生簽出時，Plastic 會檢查其是否符合任何規則，如果符合，

該路徑會處於獨佔簽出 (已鎖定)，如此一來，就沒有人

可以同時簽出。

您可以使用 'cm ^lock ^list' 以取得伺服器中的所有鎖定。

如需詳細資訊，請參閱《管理指南》：

https://www.plasticscm.com/download/help/adminguide

格式字串會將預留位置 '{0}' 取代為待簽出項目

的路徑。請查看範例，瞭解使用方法。

從 stdin 讀取輸入：

'^checkout' 命令可從 stdin 讀取路徑。若要執行此操作，可傳遞單一

破折號「-」來完成。

範例：cm ^checkout -

將讀取路徑，直到輸入空白行為止。

這可讓您使用管線以指定要簽出的檔案。

範例：

dir /S /B \*.c | cm ^checkout -

(在 Windows 中，簽出工作區中所有 .c 檔案。)

範例：

cm ^checkout file1.txt file2.txt

(簽出 'file1.txt' 和 'file2.txt' 檔案。)

cm ^co \*.txt

(簽出所有 txt 檔案。)

cm ^checkout .

(簽出目前目錄。)

cm ^checkout -^R c:\workspace\src

(遞迴簽出 'src' 資料夾。)

cm ^co file.txt --^format="正在簽出項目 {0}"

--^errorformat="簽出項目 {0} 時發生錯誤" /

--^resultformat="項目 {0} 已簽出"

(使用指定的格式字串簽出 'file.txt'，

以顯示作業的進度、結果和錯誤。)

cm ^checkout 連結 --^symlink

(簽出 '連結' 檔案，而非目標；適用於 UNIX

環境。)

cm ^checkout .-^R --^ignorefailed

(遞迴簽出目前資料夾，忽略無法簽出

的檔案。)

cm ^co .--^machinereadable --^startlineseparator=">"

(簽出目前目錄，以簡化、易於剖析的格式來列印結果，

並使用指定的字串起始行。)

== CMD\_DESCRIPTION\_CHECKSELECTORSYNTAX ==

檢查選擇器的語法。

== CMD\_USAGE\_CHECKSELECTORSYNTAX ==

用法：

cm ^checkselectorsyntax | ^css --^file=<選擇器\_檔案>

(檢查選擇器檔案語法。)

^cat <選擇器\_檔案> | cm ^checkselectorsyntax | ^css -

(Unix。從標準輸入檢查選擇器檔案。)

^type <選擇器\_檔案> | cm ^checkselectorsyntax | ^css -

(Windows。從標準輸入檢查選擇器檔案。)

--^file 要從中讀取選擇器的檔案。

== CMD\_HELP\_CHECKSELECTORSYNTAX ==

備註：

此命令會讀取檔案或標準輸入上的選擇器，並檢查其

是否為有效語法。如果語法檢查失敗，原因會

列印在標準輸出上。

範例：

cm ^checkselectorsyntax --^file=myselector.txt

(檢查 'myselector.txt' 檔案的語法。)

^cat myselector.txt | cm ^checkselectorsyntax

(從標準輸入檢查 'myselector.txt' 的語法。)

== CMD\_DESCRIPTION\_CHGREVTYPE ==

變更項目修訂類型 (二進位或文字)。

== CMD\_USAGE\_CHGREVTYPE ==

用法：

cm ^changerevisiontype | ^chgrevtype | ^crt <項目\_路徑>[ ...]--^type=(^bin | ^txt)

項目\_路徑 要變更修訂類型的項目。使用雙引號 (" ")

指定含空格的路徑。使用空格

分隔項目路徑。

--^type 目標修訂類型。選擇 '^bin' 或 '^txt'。

== CMD\_HELP\_CHGREVTYPE ==

備註：

此命令只能套用至檔案，而無法套用至目錄。

指定的類型必須是系統支援的類型：'^bin' 或 '^txt' (二進位

或文字)。

範例：

cm ^changerevisiontype c:\workspace\file.txt --^type=^txt

(將 'file.txt' 修訂類型變更為文字。)

cm ^chgrevtype comp.zip "image file.jpg" --^type=^bin

(將 'comp.zip' 和 "image file.jpg" 修訂類型變更為二進位。)

cm ^crt \*.\* --^type=^txt

(將所有檔案的修訂類型變更為文字。)

== CMD\_DESCRIPTION\_TRIGGER\_EDIT ==

編輯觸發器。

== CMD\_USAGE\_TRIGGER\_EDIT ==

用法：

cm ^trigger | ^tr ^edit <子類型\_類型> <位置\_號碼>

[--^position=<新\_位置>]

[--^name=<新\_名稱>] [--^script=<指令碼\_路徑>]

[--^filter=<字串\_篩選>] [--^server=<儲存庫伺服器規格>]

子類型\_類型 觸發器執行與觸發器作業。

輸入 'cm ^showtriggertypes' 以查看觸發器

類型的清單。

位置\_號碼 待修改之觸發器所佔的位置。

選項：

--^position 指定觸發器的新位置。

此位置不得由其他相同類型的

觸發器使用。

--^name 指定觸發器的新名稱。

--^script 指定觸發器指令碼的新執行路徑。

如果指令碼開頭為 "^webtrigger"，該指令碼將

視為 Web 觸發器。如需進一步的詳細資料，

請參閱備註。

--^filter 僅檢查符合已指定篩選條件的項目。

--^server 在指定的伺服器上修改觸發器。

如果未指定任何伺服器，請在用戶端上設定的

伺服器上執行命令。

(使用 'cm ^help ^objectspec' 以深入瞭解伺服器

規格。)

== CMD\_HELP\_TRIGGER\_EDIT ==

備註：

Web 觸發器：藉由輸入 "^webtrigger <目標-uri>"

做為觸發器命令，以建立 Web 觸發器。在此情況下，觸發器將對應指定的 URI 來執行 POST 查詢，

其中要求本文包含具有觸發器環境變數的 JSON

字典，以及

指向字串陣列的 INPUT 金鑰。

範例：

cm ^trigger ^edit ^after-setselector 6 --^name="Backup2 管理員" --^script="/new/path/al/script"

cm ^tr ^edit ^before-mklabel 7 --^position=4 --^server=myserver:8084

cm ^trigger ^edit ^after-add 2 --^script="^webtrigger http://myserver.org/api"

== CMD\_DESCRIPTION\_CODEREVIEW ==

建立、編輯或刪除程式碼檢閱。

== CMD\_USAGE\_CODEREVIEW ==

用法：

cm ^codereview <規格> <標題> [--^status=<狀態\_名稱>]

[--^assignee=<使用者\_名稱>] [--^format=<字串\_格式>]

[--^repository=<儲存庫\_規格>]

(建立程式碼檢閱。)

cm ^codereview -^e <ID> [--^status=<狀態\_名稱>] [--^assignee=<使用者\_名稱>]

[--^repository=<儲存庫\_規格>]

(編輯程式碼檢閱。)

cm ^codereview -^d <ID> [ ...][--^repository=<儲存庫\_規格>]

(刪除一或多個程式碼檢閱。)

規格 規格可能是變更集規格或分支規格。

該規格將會是新程式碼檢閱的目標。(使用

'cm ^help ^objectspec' 以深入瞭解變更集或

分支規格。)

標題 做為新

程式碼檢閱標題的文字字串。

ID 程式碼檢閱識別碼。也可以使用

GUID。

選項：

-^e 編輯現有程式碼檢閱的參數。

-^d 刪除一或多個現有的程式碼檢閱。使用

空格來分隔程式碼檢閱 ID。

--^status 設定程式碼檢閱的新狀態。如需其他資訊，

請參閱備註。

--^assignee 設定程式碼檢閱新的受指派者。

--^format 以指定的格式擷取輸出訊息。如需

其他資訊，請參閱備註。

--^repository 設定要做為預設使用的儲存庫。(使用

'cm ^help ^objectspec' 以深入瞭解儲存庫

規格。)

== CMD\_HELP\_CODEREVIEW ==

備註：

此命令可讓使用者管理程式碼檢閱：建立、編輯和刪除

變更集或分支的程式碼檢閱。

若要建立新的程式碼檢閱，需要變更集/分支規格

和標題。也可以設定初始狀態和受指派者。結果會傳回 ID

(或 GUID，如果有的話)。

若要編輯或刪除現有的程式碼檢閱，需要有目標程式碼檢閱 ID

(或 GUID)。如果沒有錯誤，則不會顯示任何訊息。

狀態參數只能是以下其中之一：^"Under review"

(預設)、^"Reviewed" 或 ^"Rework required"。

儲存庫參數可用於設定預設的工作中

儲存庫。當使用者想要在與目前工作區相關的伺服器

不同的伺服器上管理檢閱時，或完全沒有

目前工作區時，這相當有用。

輸出格式自訂化：

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{0} ID

{1} GUID

請注意，'--^format' 參數只有在建立

新的程式碼檢閱時才會生效。

範例：

cm ^codereview ^cs:1856@myrepo@myserver:8084 "我的程式碼檢閱" --^assignee=虛設

cm ^codereview ^br:/main/task001@myrepo@myserver:8084 "我的程式碼檢閱" \

--^status=^"Rework required" --^assignee=新人 --^format="{^id} -> {^guid}"

cm ^codereview 1367 -^e --^assignee=新\_受指派者

cm ^codereview -^e 27658884-5dcc-49b7-b0ef-a5760ae740a3 --^status=已檢閱

cm ^codereview -^d 1367 --^repository=myremoterepo@myremoteserver:18084

cm ^codereview 27658884-5dcc-49b7-b0ef-a5760ae740a3 -^d

== CMD\_DESCRIPTION\_CRYPT ==

加密密碼。

== CMD\_USAGE\_CRYPT ==

用法：

cm ^crypt <我的密碼>

我的密碼 要加密的密碼。

== CMD\_HELP\_CRYPT ==

備註：

此命令會對傳遞做為引數的指定密碼進行加密。

其設計目的是加密設定檔案中的密碼，並提高

安全性。

範例：

cm ^crypt dbconfpassword -> ENCRYPTED: encrypteddbconfpassword

(加密資料庫設定檔案中的密碼：'db.conf'。)

== CMD\_DESCRIPTION\_DEACTIVATEUSER ==

停用授權使用者。

== CMD\_USAGE\_DEACTIVATEUSER ==

用法：

cm ^deactivateuser | ^du <使用者\_名稱>[ ...][--^server=<名稱:連接埠>]

[--^nosolveuser]

使用者\_名稱 要停用的使用者名稱。使用空格以分隔

使用者名稱。

如果是 SID，則需要 '--^nosolveuser'。

選項：

--^server 在指定的伺服器中停用使用者。

如果未指定任何伺服器，請在用戶端上設定的

伺服器上執行命令。

--^nosolveuser 使用此選項，命令將不會檢查使用者名稱

是否存在於驗證系統中。此

<使用者\_名稱> 必須是使用者 SID。

== CMD\_HELP\_DEACTIVATEUSER ==

備註：

此命令會將使用者設為非作用中，禁止該使用者使用

Plastic SCM。

如需啟動 Plastic SCM 使用者的詳細資訊，請參閱 'cm ^activateuser'

命令。

此命令會檢查使用者是否存在於基礎驗證系統 (例如，

ActiveDirectory、LDAP、使用者/密碼...)。

若要強制停用驗證系統中不再存在的使用者，

您可以使用 '--^nosolveuser' 選項。

範例：

cm ^deactivateuser john

cm ^du peter "mary collins"

cm ^deactivateuser john --^server=myserver:8084

cm ^deactivateuser S-1-5-21-3631250224-3045023395-1892523819-1107 --^nosolveuser

== CMD\_DESCRIPTION\_DIFF ==

顯示檔案、變更集和標籤之間的差異。

== CMD\_USAGE\_DIFF ==

用法：

cm ^diff <變更集規格> | <標籤規格> | <擱置規格> [<變更集規格> | <標籤規格> | <擱置規格>]

[<路徑>]

[--^added] [--^changed] [--^moved] [--^deleted]

[--^repositorypaths] [--^download=<下載\_路徑>]

[--^encoding=<名稱>]

[--^ignore=(^eol | ^whitespaces | ^"eol&whitespaces" | ^none)]

[--^clean]

[--^format=<字串\_格式>] [--^dateformat=<字串\_格式>]

顯示「來源」變更集或擱置集與

「目的地」變更集或擱置集之間的差異。可使用變更集或標籤規格

來指定變更集。

在提供兩種規格的情況下，第一個是差異的「來源」；

第二個是「目的地」。

在僅提供一種規格的情況下，「來源」將會是指定

「目的地」的上層變更集。

如果已指定選用的路徑，[差異比對] 視窗將會啟動，以顯示

該檔案兩種修訂之間的差異。

cm ^diff <修訂規格1> <修訂規格2>

顯示一對修訂之間的差異。差異顯示在

[差異比對] 視窗中。指定的第一個修訂將出現

在左側。

cm ^diff <分支規格> [--^added] [--^changed] [--^moved] [--^deleted]

[--^repositorypaths] [--^download=<下載\_路徑>]

[--^encoding=<名稱>]

[--^ignore=(^eol | ^whitespaces | ^"eol&whitespaces" | ^none)]

[--^clean]

[--^format=<字串\_格式>] [--^dateformat=<字串\_格式>]

[--^fullpaths | --^fp]

顯示分支差異。

(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

選項：

--^added 僅列印包含新增至儲存庫之

項目的差異。

--^changed 僅列印包含變更項目

的差異。

--^moved 僅列印包含已移動或已重新命名

項目的差異。

--^deleted 僅列印包含已刪除項目

的差異。

如果未指定 '--^added'、'--^changed'、'--^moved' 或 '--^deleted'，

則此命令會列印所有差異。

'^A' 表示已新增項目。

'^C' 表示已變更項目。

'^D' 表示已刪除項目。

'^M' 表示已移動項目。左側項目是原始的，

右側項目是目的地。

--^repositorypaths 列印儲存庫路徑，而非工作區路徑。

(此選項會覆寫 '--^fullpaths' 選項。)

--^download 儲存指定的輸出路徑中的

差異內容。

--^encoding 指定輸出編碼，例如：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

--^ignore 設定指定的比較方法。

如需詳細資訊，請參閱備註。

--^clean 不考慮因合併而產生的差異，

僅考慮簡單簽入所建立

的差異。

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

--^dateformat 用於輸出日期的格式。

--^fullpaths, --^fp 如果可能，會強制列印檔案和目錄的

完整工作區路徑。

== CMD\_HELP\_DIFF ==

備註：

比較方法:

^eol 忽略行尾差異。

^whitespaces 忽略空格差異。

^"eol&whitespaces" 忽略行尾和空格差異。

^none 偵測行尾和空格差異。

此命令接受格式字串以顯示輸出。

此命令的參數如下所示：

{^path} 項目路徑。

{^date} 變更日期/時間。

{^owner} 變更作者。

{^revid} 差異中視為目的地之修訂的

修訂 ID。

{^parentrevid} 差異中視為目的地之修訂上層的

修訂 ID。

{^baserevid} 差異中視為來源之修訂的

修訂 ID。

{^srccmpath} 移動項目 (移動作業) 之前的伺服器路徑。

{^dstcmpath} 移動項目 (移動作業) 之後的伺服器路徑。

{^type} 項目類型：

^D 目錄、

^B 二進位檔案、

^F 文字檔、

^S 符號連結、

^X Xlink。

{^repository} 項目的儲存庫。

{^status} 項目狀態：

^A 已新增、

^D 已刪除、

^M 已移動、

^C 已變更。

{^fsprotection} 顯示項目權限 (Linux/Mac chmod)。

{^srcfsprotection} 顯示上層修訂項目權限。

{^newline} 插入新行。

'^revid' 的說明：

對於已新增的項目，'^baserevid' 和 '^parentrevid' 將會是 -1，因為

在此情況下，不存在任何先前修訂。

對於已刪除的項目，'^revid' 是來源修訂的 ID，而且

'^baserevid' 將會是 -1，因為沒有任何目的地修訂。

對於 Xlinks，'^baserevid' 和 '^parentrevid' 一律為 -1。

範例：

比較分支：

cm ^diff ^br:/main/task001

cm ^diff ^br:/main/task001 \doc\readme.txt

比較變更集樹狀目錄：

cm ^diff 19

cm ^diff 19 25

cm ^diff ^cs:19 ^cs:25 --^format="{^path} {^parentrevid}"

cm ^diff ^cs:19 ^cs:23 --^format="{^date} {^path}" --^dateformat="yy/dd/MM HH:mm:ss"

cm ^diff ^cs:19 ^cs:23 --^changed

cm ^diff ^cs:19 ^cs:23 --^repositorypaths

cm ^diff ^cs:19 ^cs:23 --^download="D:\temp"

cm ^diff ^cs:19 ^cs:23 --^clean

cm ^diff ^cs:19 ^cs:23 \doc\readme.txt

比較標籤樹狀目錄：

cm ^diff ^lb:FirstReleaseLabel ^lb:SecondReleaseLabel

cm ^diff ^lb:tag\_193.2 ^cs:34214

cm ^diff ^cs:31492 ^lb:tag\_193.2

比較擱置樹狀目錄：

cm ^diff ^sh:2

cm ^diff ^sh:2 ^sh:4

比較修訂規格：

cm ^diff ^rev:readme.txt#^cs:19 ^rev:readme.txt#^cs:20

cm ^diff ^serverpath:/doc/readme.txt#^cs:19@myrepo \

^serverpath:/doc/readme.txt#^br:/main@myrepo@localhost:8084

== CMD\_DESCRIPTION\_DIFFMETRICS ==

顯示兩個修訂之間的不同指標。

== CMD\_USAGE\_DIFFMETRICS ==

用法：

cm ^diffmetrics | ^dm <修訂規格1> <修訂規格2> [--^format=<字串\_格式>]

[--^encoding=<名稱>]

[--^ignore=(^eol | ^whitespaces | ^"eol&whitespaces" | ^none)]

修訂規格 用於比較的修訂。

(使用 'cm ^help ^objectspec' 以深入瞭解修訂規格。)

選項：

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

--^encoding 指定輸出編碼，例如：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

--^ignore 設定指定的比較方法。

如需詳細資訊，請參閱備註。

== CMD\_HELP\_DIFFMETRICS ==

備註：

指標為：已變更、已新增和已刪除的行數。

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{0} 已變更的行數。

{1} 已新增的行數。

{2} 已刪除的行數。

範例：

cm ^diffmetrics file.txt#^cs:2 file.txt#^br:/main/scm0211 \

--^format="有 {0} 個已變更、{1} 個已新增和 {2} 已刪除的行。"

(擷取格式化的 diffmetrics 結果。)

cm ^dm file.txt#^cs:2 file.txt#^cs:3 --^encoding=utf-8 --^ignore=^whitespaces

== CMD\_DESCRIPTION\_FASTEXPORT ==

以快速匯出格式匯出儲存庫。

== CMD\_USAGE\_FASTEXPORT ==

用法：

cm ^fast-export | ^fe <儲存庫規格> <快速匯出檔案>

[--^import-marks=<標記\_檔案>]

[--^export-marks=<標記\_檔案>]

[--^branchseparator=<字元\_分隔符號>]

[--^nodata] [--^from=<變更集 ID>] [--^to=<變更集 ID>]

選項:

儲存庫規格 將從中匯出資料的儲存庫。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫規格。)

快速匯出檔案 內含儲存庫資料的檔案，格式為 Git

快速匯出。

--^import-marks 用於增量匯入的標記檔案。此檔案

先前已由 '--^export-marks' 匯出。將不會匯入此檔案中

描述的變更集，

因為這些變更集先前已匯入。

--^export-marks 將儲存已匯入變更集的檔案。

此檔案用於之後進行的快速匯入，向已匯入的變更集

發出訊號。

--^branchseparator Plastic SCM 使用「/」做為分支階層中的預設

分隔符號。此選項允許使用字元做為階層分隔符號，

因此 main-task-sub 在 Plastic SCM 中會描繪成

/main/task/sub。

--^nodata 匯出儲存庫但不包含資料。此命令

對於檢查匯出是否將正常執行非常有用。

--^from 從特定變更集匯出。

--^to 匯出至特定變更集。

== CMD\_HELP\_FASTEXPORT ==

備註：

- 若要將 Plastic SCM 儲存庫匯入至 Git，請使用命令，例如：

^cat repo.fe.00 | ^git ^fast-import --^export-marks=marks.git --^import-marks=marks.git

- 支援使用包含先前匯入之變更集的標記檔案

('--^import-marks' 和 '--^export-marks'

檔案) 進行增量匯出。

這表示只有在先前快速匯出中未匯出的

新變更集才會匯出。

範例：

cm ^fast-export repo@localhost:8087 repo.fe.00 --^import-marks=marks.cm \

--^export-marks=marks.cm

(將本地伺服器中的儲存庫 'repo' 匯出至 Git 快速匯出格式的

'repo.fe.00' 檔案，並建立標記檔案，以便之後執行

增量匯出。)

cm ^fast-export repo@localhost:8087 repo.fe.00 --^from=20

(從變更集 '20' 將本地伺服器中的儲存庫 'repo' 以 Git 快速匯出格式

匯出至 'repo.fe.00'檔案。)

== CMD\_DESCRIPTION\_FASTIMPORT ==

將 Git 快速匯出資料匯入儲存庫。

== CMD\_USAGE\_FASTIMPORT ==

用法：

cm ^fast-import | ^fi <儲存庫規格> <快速匯出檔案>

[--^import-marks=<標記\_檔案>]

[--^export-marks=<標記\_檔案>]

[--^stats] [--^branchseparator=<字元\_分隔符號>]

[--^nodata] [--^ignoremissingchangesets] [--^mastertomain]

選項：

儲存庫規格 將匯入資料的

儲存庫。如果儲存庫先前

不存在，則會建立一個儲存庫。(使用 'cm ^help ^objectspec' 以深入瞭解

儲存庫規格。)

快速匯出檔案 內含儲存庫資料的檔案，格式為 Git

快速匯出。

--^import-marks 用於增量匯入的標記檔案。

此檔案

先前已由 '--^export-marks' 匯出。將不會匯入此檔案中

描述的變更集，

因為這些變更集先前已匯入。

--^export-marks 將儲存已匯入變更集的

檔案。此檔案用於之後進行的

快速匯入，向已匯入的

變更集發出訊號。

--^stats 列印關於匯入過程的一些統計資料。

--^branchseparator Plastic SCM 使用「/」做為分支階層中的預設

分隔符號。此選項允許使用

字元做為階層分隔符號，因此 main-task-sub

在 Plastic SCM 中會描繪成 /main/task/sub。

--^nodata 匯入 Git 快速匯出但不包含

資料。此命令

對於檢查匯入是否將正常執行非常有用。

--^ignoremissingchangesets 會捨棄無法匯入的

任何變更集，且快速匯入作業

會在無變更集的情況下繼續進行。

--^mastertomain 使用 "^main" 匯入，而非使用 "^master"。

== CMD\_HELP\_FASTIMPORT ==

備註：

- 若要匯出 git 儲存庫，請使用命令，例如：

^git ^fast-export --^all -^M --^signed-tags=^strip --^tag-of-filtered-object=^drop> ..\git-fast-export.dat

-^M 選項對於偵測移動的項目很重要。

- 如果指定的儲存庫不存在，則會建立該儲存庫。

- 支援使用包含

先前匯入之變更集的標記檔案 ('--^import-marks' 和 '--^export-marks'

檔案) 進行增量匯入。

這表示只有在

先前快速匯入中未匯入的新變更集才會匯入。

範例：

cm ^fast-import mynewrepo@atenea:8084 repo.fast-export

(將 'repo.fast-export' 檔案中匯出的內容匯入伺服器

'atenea:8084' 上的 'mynewrepo' 儲存庫。)

cm ^fast-import repo@atenea:8084 repo.fast-export --^export-marks=rep.marks

(將 'repo.fast-export' 檔案中匯出的內容匯入伺服器

'atenea:8084' 上的 'repo' 儲存庫，並建立標記檔案，

以便之後執行增量匯入。)

cm ^fast-import repo@server:8084 repo.fast-export --^import-marks=repo.marks \

--^export-marks=repo.marks

(匯入 'repo.fast-export' 檔案的內容。只有不在標記

檔案中的新變更集才會匯入。相同的標記

檔案用於再次儲存變更集的清單，以便下次進行

增量匯入。)

== CMD\_DESCRIPTION\_FILEINFO ==

擷取工作區中項目的詳細資訊。

== CMD\_USAGE\_FILEINFO ==

用法：

cm ^fileinfo <項目\_路徑>[ ...][--^fields=<欄位\_值>[,...]]

[[--^xml | -^x [=<輸出\_檔案>]] | [--^format=<字串\_格式>]]

[--^symlink] [--^encoding=<名稱>]

項目\_路徑 要顯示的項目。使用空格以分隔

項目。

使用雙引號 (" ") 指定含空格的

路徑。

選項：

--^fields 以逗號分隔值的字串。此選項會選取要

針對每個項目列印的欄位。如需詳細資訊，

請參閱備註。

--^xml | -^x 將 XML 格式的輸出列印至標準輸出。

可以指定輸出檔案。此選項

無法與 '--^format' 合併。

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。此選項無法

與 '--^xml' 合併。

如果同時指定兩個選項，則此 '--^format' 選項

優先於 '--^fields'。

--^symlink 將檔案資訊作業套用至符號連結，而不是

套用至目標。

--^encoding 指定輸出編碼，例如：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

== CMD\_HELP\_FILEINFO ==

備註：

此命令會列印每個選取項目之屬性的詳細清單。

預設情況下，每個屬性會列印在新行上。

屬性清單可修改為僅顯示使用者需要的

屬性。這可以使用 '--^fields=<欄位\_清單>' 來達成，其中接受

一連串以逗號分隔的屬性名稱。如此一來，只會顯示已指定

名稱的引數。

修訂前端變更集：

預設情況下，已停用此選項。請注意，擷取這個屬性

的速度明顯比其他屬性慢，因此，我們建議使用者

將盡可能多的項目分組在一起。這可避免執行許多

個別的 'cm ^fileinfo'，以改善執行時間。

此外，此功能目前無法用於受控目錄。

您可以在下方尋找可用屬性名稱的完整清單。

預設情況下，不會顯示標記星號 ('\*') 的名稱：

^ClientPath 該項目在磁碟上的本地路徑。

^RelativePath 工作區相對路徑。

^ServerPath 該項目的儲存庫路徑。

(請注意：此選項目前不支援

轉換的工作區)。

^Size 項目大小。

^Hash 項目雜湊加總。

^Owner 項目所屬的使用者。

^RevisionHeadChangeset (\*) 在分支的前端變更集中

載入的修訂變更集。

(請參閱下方備註。)

^RevisionChangeset 工作區中目前載入之修訂

的變更集。

^RepSpec 該項目的儲存庫規格。

(使用 'cm ^help ^objectspec' 以深入瞭解

儲存庫規格。)

^Status 工作區項目狀態：已新增、已簽出、

已刪除等。

^Type 修訂類型 (文字、二進位、目錄、符號連結

或未知)。

^Changelist 項目所屬的變更集 (如果有的話)。

^IsLocked (\*) 項目是否受到獨佔簽出的

鎖定。

^LockedBy (\*) 獨佔簽出項目的使用者。

^LockedWhere (\*) 獨佔簽出項目

的位置。

^IsUnderXlink 項目是否位於 Xlink

下方。

^UnderXlinkTarget 項目所在的 Xlink 目標

(如果有的話)。

^UnderXlinkPath Xlink 連結之儲存庫中的項目伺服器路徑

(如果有的話)。

^UnderXlinkWritable 項目所屬的 Xlink 是否

可寫入。

^UnderXlinkRelative 項目所屬的 Xlink 是否

相關。

^IsXlink 項目本身是否為 Xlink。

^XlinkTarget 如果項目是 Xlink，該項目指向的目標

儲存庫。

^XlinkName 項目的 Xlink 名稱，如果實際

存在的話。

^XlinkWritable Xlink 項目是否為可寫入的

Xlink。

^XlinkRelative Xlink 項目是否為相關的

Xlink。

輸出格式自訂化：

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{^ClientPath}

{^RelativePath}

{^ServerPath}

{^Size}

{^Hash}

{^Owner}

{^RevisionHeadChangeset}

{^RevisionChangeset}

{^Status}

{^Type}

{^Changelist}

{^IsLocked}

{^LockedBy}

{^LockedWhere}

{^IsUnderXlink}

{^UnderXlinkTarget}

{^UnderXlinkPath}

{^UnderXlinkWritable}

{^UnderXlinkRelative}

{^IsXlink}

{^XlinkTarget}

{^XlinkName}

{^XlinkWritable}

{^XlinkRelative}

{^RepSpec}

請注意，'--^format' 和 '--^xml' 選項是互斥的，因此，

無法同時使用這兩個選項。

範例：

cm ^fileinfo file1.txt file2.txt dir/

cm ^fileinfo "New Project.csproj" --^xml

cm ^fileinfo assets.art --^fields=^ServerPath,^Size,^IsLocked,^LockedBy

cm ^fileinfo proj\_specs.docx --^fields=^ServerPath,^RevisionChangeset --^xml

cm ^fileinfo samples.ogg --^format="{^ServerPath}[{^Owner}] -> {^Size}"

== CMD\_DESCRIPTION\_FIND\_QUERY ==

執行類似 SQL 的查詢，以尋找 Plastic SCM 物件。

== CMD\_USAGE\_FIND\_QUERY ==

用法：

cm ^find <物件\_類型>

[^where <字串\_條件>]

[^on ^repository '<儲存庫規格>' | ^on ^repositories '<儲存庫規格1>','<儲存庫規格2>'[,...]]

[--^format=<字串\_格式>] [--^dateformat=<日期\_格式>]

[--^nototal] [--^file=<傾印\_檔案>] [--^xml]

[--^encoding=<名稱>]

物件\_類型 要尋找的物件類型。

使用 'cm ^help ^showfindobjects' 以瞭解如何指定

這些物件。

您也可以參閱 'cm ^find' 指南：

https://www.plasticscm.com/download/help/cmfind

選項：

字串\_條件 物件屬性上的搜尋條件。

儲存庫規格 搜尋儲存庫別名或規格。

在 '^on ^repositories' 的情況下，使用逗號以

分隔儲存庫規格欄位。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫

規格。)

--^format 以指定的格式擷取輸出訊息。

請參閱 'cm ^find' 指南，以瞭解可做為

輸出格式字串的所有物件屬性：

https://www.plasticscm.com/download/help/cmfind

--^dateformat 用於輸出日期的格式。

--^nototal 結尾不輸出記錄計數。

--^file 要傾印結果的檔案。

--^xml 將 XML 格式的輸出列印至標準輸出。

--^encoding 指定輸出編碼，例如：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

== CMD\_HELP\_FIND\_QUERY ==

備註：

如果未指定任何儲存庫，將會在工作區中設定的

儲存庫上進行搜尋。

當您從命令列使用比較運算子 (>、<、>=、<=) 執行查詢時，

請記住，此命令介面會將這些運算子視為 IO

重新導向。因此，您需要使用雙引號括住

查詢。

'cm ^find' 命令接受格式字串以顯示輸出。

每個輸出參數是根據字串加以識別，而且使用者可以藉由在

'{' 和 '}' 括號之間輸入參數號碼來引用參數。

輸出參數通常對應至物件的屬性。

這些是一些有效的輸出格式字串：

--^format={^id}{^date}{^name}

--^format="{^item}#{^branch} ^with ^date {^date}"

XML 與編碼考量：

指定 '--^xml' 選項時，命令會將命令結果以 XML 文字的形式

顯示在標準輸出中。會使用作業系統預設編碼來顯示文字，

因此，在主控台中可能會不正確地顯示

非 ANSI 字元。如果您將命令輸出重新導向至檔案，

將會正確顯示。當同時指定 '--^xml' 和 '--^file'

選項時，預設編碼將會是 utf-8。

範例：

cm ^find ^revision

cm ^find ^revision "^where ^changeset=23 ^and ^owner='maria'"

cm ^find ^branch "^on ^repository 'rep1'"

cm ^find ^label "^on ^repositories 'rep1', '^rep:default@localhost:8084'"

cm ^find ^branch "^where ^parent='^br:/main' ^on ^repository 'rep1'"

cm ^find ^revision "^where ^item='^item:.'" --^format="{^item}#{^branch}"

cm ^find ^revision "^where ^item='^item:.'" --^xml --^file=c:\queryresults\revs.xml

== CMD\_DESCRIPTION\_FINDCHANGED ==

取得已變更檔案的清單。此命令已棄用，並且單純為了

回溯相容性而保留。請改用 'cm ^status'。

== CMD\_USAGE\_FINDCHANGED ==

用法：

cm ^findchanged | ^fc [-^R | -^r | --^recursive] [--^checkcontent]

[--^onlychanged] [<路徑>]

選項：

-^R 在目錄中遞迴尋找。

--^checkcontent 依內容比較檔案。

--^onlychanged 僅尋找已變更的檔案；不會獲得

簽出。

路徑 (預設：目前目錄。)

尋找已變更檔案的初始路徑。

== CMD\_HELP\_FINDCHANGED ==

備註：

如果未提供任何 '--^checkcontent' 選項，Plastic 會根據檔案時間戳記

尋找變更。指定 '--^checkcontent' 選項時，會比較檔案或

資料夾內容，而不是使用時間戳記。

此命令對於在與 Plastic SCM 伺服器中斷連線時偵測已變更的檔案，

相當有用。此輸出可透過管道傳送至簽出命令，

在稍後檢查變更 (請參閱範例)。

範例：

cm ^findchanged .

(在目前目錄中尋找已變更的檔案。)

cm ^findchanged -^R .| cm ^checkout -

(簽出已變更的元素。)

== CMD\_DESCRIPTION\_FINDCHECKEDOUT ==

取得已簽出項目的清單。此命令已棄用，並且單純為了

回溯相容性而保留。請改用 'cm ^status'。

== CMD\_USAGE\_FINDCHECKEDOUT ==

用法：

cm ^findcheckouts | ^fco [--^format=<字串\_格式>] [--^basepath]

選項：

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

--^basepath 開始搜尋簽出的路徑。如果

未指定，會使用目前路徑。

== CMD\_HELP\_FINDCHECKEDOUT ==

備註：

此命令對於在單一步驟中簽入或取消簽出所有已簽出項目相當有用，

可將標準輸出重新導向其他命令。

請參閱範例。

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{0} 日期。

{1} 擁有者。

{2} 工作空間資訊。

{3} 用戶端機器名稱。

{4} 項目路徑。

{5} 分支和儲存庫資訊。

範例：

cm ^findcheckouts --^format="在分支 {5} 上已變更檔案 {4} "

(尋找已簽出項目，並使用檔案路徑和分支

與儲存庫資訊來設定輸出的格式。)

cm ^findcheckouts --^format={4} | cm ^checkin -

(簽入所有已簽出項目。)

cm ^findcheckouts --^format={4} | cm ^undocheckout -

(取消簽出所有已簽出項目。)

== CMD\_DESCRIPTION\_FINDPRIVATE ==

取得私人項目的清單。此命令已棄用，並且單純為了

回溯相容性而保留。請改用 'cm ^status'。

== CMD\_USAGE\_FINDPRIVATE ==

用法：

cm ^findprivate | ^fp [-^R | -^r | --^recursive] [--^exclusions] [<路徑>]

選項：

-^R 在目錄中遞迴尋找。

--^exclusions 此選項允許在忽略路徑內剪下搜尋，

由檔案 ignore.conf 定義。

路徑 (預設：目前目錄。)

尋找私人檔案的初始路徑。

== CMD\_HELP\_FINDPRIVATE ==

備註：

如果有指定任何路徑，Plastic SCM 將開始從

目前目錄搜尋。

此命令對於在資料夾上新增私人項目相當有用，透過管道將輸出

傳送至新增命令。請參閱範例。

範例：

cm ^findprivate .

cm ^findprivate -^R | cm ^add -

(遞迴搜尋私人項目並新增這些項目。)

== CMD\_DESCRIPTION\_GETCONFIG ==

獲得設定資訊。

== CMD\_USAGE\_GETCONFIG ==

用法：

cm ^getconfig [^setfileasreadonly] [^location] [^extensionworkingmode]

[^extensionprefix] [^defaultrepserver]

^setfileasreadonly 傳回受保護的檔案是否保持

唯讀。

^location 傳回用戶端設定路徑。

^extensionworkingmode 傳回副檔名工作模式。

^extensionprefix 傳回設定的副檔名前置詞。

^defaultrepserver 傳回預設儲存庫伺服器的

位置。

== CMD\_HELP\_GETCONFIG ==

範例：

cm ^getconfig ^setfileasreadonly

== CMD\_DESCRIPTION\_GETFILE ==

下載指定修訂的內容。

== CMD\_USAGE\_GETFILE ==

用法：

cm ^getfile | ^cat <修訂規格> [--^file=<輸出\_檔案>] [--^debug]

[--^symlink] [--^raw]

修訂規格 物件規格。(使用 'cm ^help ^objectspec' 以深入瞭解

規格。)

選項：

--^file 儲存輸出的檔案。預設情況下，該檔案會列印在

標準輸出上。

--^debug 使用目錄規格時，此命令會顯示目錄中的

所有項目、項目的修訂 ID，

以及檔案系統保護。

--^symlink 將作業套用至符號連結，而不是套用至

目標。

--^raw 顯示檔案的未經處理資料。

== CMD\_HELP\_GETFILE ==

範例：

cm ^cat myfile.txt#^br:/main

(在 'myfile.txt' 的分支 '^br:/main' 中，獲取最後一個修訂。)

cm ^getfile myfile.txt#^cs:3 --^file=tmp.txt

(獲取 'myfile.txt' 的變更集 3，並將其寫入檔案 'tmp.txt'。)

cm ^cat ^serverpath:/src/foo.c#^br:/main/task003@myrepo

(獲取儲存庫 'myrepo' 中分支 '/main/task003' 的最後

一個變更集 '/src/foo.c' 的內容)

cm ^cat ^revid:1230@^rep:myrep@^repserver:myserver:8084

(獲取 ID 1230 的修訂。)

cm ^getfile ^rev:info\ --^debug

(獲取 'info' 儲存庫中的所有修訂。)

== CMD\_DESCRIPTION\_GETREVISION ==

在工作區中載入修訂。

== CMD\_USAGE\_GETREVISION ==

此命令會修改工作區中載入的修訂，因此，會影響

未來合併。

這是繼承自舊版本的進階命令，所以請謹慎使用。

用法：

cm ^getrevision <修訂規格>

修訂規格 物件規格。(使用 'cm ^help ^objectspec' 以深入瞭解

修訂規格。)

== CMD\_HELP\_GETREVISION ==

範例：

cm ^getrevision file.txt#^cs:3

(取得 'file.txt' 的變更集 3 修訂。)

== CMD\_DESCRIPTION\_GETSTATUS ==

取得項目的狀態。

== CMD\_USAGE\_GETSTATUS ==

這是自動化命令，目的僅限用於自動化 'cm'。

方便使用的程度並不高。

用法：

cm ^getstatus | ^gs <項目\_路徑>[ ...][--^format=<字串\_格式>] [--^stats]

[-^R | -^r | --^recursive]

項目\_路徑 要取得狀態的一或多個項目。使用雙引號

(" ") 指定含空格的路徑。使用

空格分隔路徑。

選項：

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

--^stats 列印關於取得狀態過程的一些統計資料。

-^R 在目錄中遞迴顯示狀態。

== CMD\_HELP\_GETSTATUS ==

備註：

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{0} 項目路徑。

{1} 項目狀態：

0 私人，

1 已簽入，

2 已簽出。

從 stdin 讀取輸入：

'^getstatus' 命令可從 stdin 讀取路徑。若要執行此操作，可

傳遞單一破折號「-」來完成。

範例：cm ^getstatus -

將讀取路徑，直到輸入空白行為止。

這可讓您使用管線以指定要取得狀態的路徑。

範例：

dir /S /B \*.c | cm ^getstatus --^format="路徑 {0} 狀態 {1}" -

(在 Windows 中，取得工作區中所有 .c 檔案的狀態。)

範例：

cm ^getstatus file1.txt file2.txt

(取得檔案的狀態。)

cm ^gs info\ -^R --^format="項目 {0} 具有狀態 {1}"

(取得目錄及其項目的狀態，並顯示

已格式化的輸出。)

== CMD\_DESCRIPTION\_GETTASKBRANCHES ==

取得與任務連結的分支。

== CMD\_USAGE\_GETTASKBRANCHES ==

這是自動化命令，目的僅限用於自動化 'cm'。

方便使用的程度並不高。

用法：

cm ^gettaskbranches | ^gtb <任務\_名稱> [--^format=<字串\_格式>]

[--^dateformat=<日期\_格式>]

任務\_名稱 任務識別碼。

選項：

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

--^dateformat 用於輸出日期的格式。

== CMD\_HELP\_GETTASKBRANCHES ==

備註：

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{^tab} 插入定位字元空格。

{^newline} 插入新行。

{^name} 分支名稱。

{^owner} 分支的擁有者。

{^date} 建立分支的日期。

{^type} 分支類型 (如果是智慧型，則為 'T'，如果不是，則為 'F')。

{^parent} 上層分支。

{^comment} 分支的註解。

{^repname} 存在此分支的儲存庫。

{^repserver} 伺服器名稱。

範例：

cm ^gettaskbranches 4311

cm ^gtb 4311 --^format="^br:{^name}"

cm ^gtb 4311 --^format="^br:{^name} {^date}" --^dateformat="yyyy/MM/dd HH:mm:ss"

== CMD\_DESCRIPTION\_GETWWI ==

顯示關於工作區選擇器的資訊。

== CMD\_USAGE\_GETWWI ==

用法：

cm ^wi [<工作區\_路徑>]

選項：

工作區\_路徑 機器上的工作區路徑。

== CMD\_HELP\_GETWWI ==

備註：

'^wi' 命令會顯示工作區 (儲存庫、分支，及/或標籤)

的作用中設定。

範例：

cm ^wi c:\mywk

== CMD\_DESCRIPTION\_GWP ==

從路徑取得工作區資訊。

== CMD\_USAGE\_GWP ==

這是自動化命令，目的僅限用於自動化 'cm'。

方便使用的程度並不高。

用法：

cm ^getworkspacefrompath | ^gwp <項目\_路徑> [--^format=<字串\_格式>]

項目\_路徑 磁碟上的檔案或資料夾。

選項：

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

== CMD\_HELP\_GWP ==

備註：

此命令會顯示位於路徑中的工作區相關資訊。

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{0} | {^wkname} 工作區名稱。

{1} | {^wkpath} 工作區路徑。

{2} | {^machine} 用戶端機器名稱。

{3} | {^owner} 工作區擁有者。

{4} | {^guid} 工作區 GUID。

{^tab} 插入定位字元空格。

{^newline} 插入新行。

範例：

cm ^gwp c:\myworkspace\code\file1.cpp --^format="工作區名稱: {^wkname}"

== CMD\_DESCRIPTION\_HELP ==

取得 Plastic SCM 命令的說明。

== CMD\_USAGE\_HELP ==

用法：

cm ^help <命令>

== CMD\_HELP\_HELP ==

== CMD\_DESCRIPTION\_IOSTATS ==

顯示關於硬體的統計資料。

== CMD\_USAGE\_IOSTATS ==

用法：

cm ^iostats [<儲存庫伺服器規格>] [<測試\_清單>[ ...]]

[--^nettotalmb=<值\_mb>] [--^networkiterations=<值\_反覆項目>]

[--^diskdatasize=<值\_大小>] [--^disktestpath=<值\_路徑>]

[--^systemdisplaytime=<值\_時間>]

[--^systemdisplaytimeinterval=<值\_間隔>]

選項：

儲存庫伺服器規格 執行網路測試的可用 Plastic SCM

伺服器，例如 "serverUploadTest"

及/或 "serverDownloadTest"。

如果未提供任何伺服器，預設情況下，

此命令會嘗試與設定的伺服器

進行通訊。

(使用 'cm ^help ^objectspec' 以深入瞭解

伺服器規格。)

測試\_清單 可用測試。使用空格分隔

測試欄位。

如需詳細資訊，請參閱備註。

--^nettotalmb 表示網路測試傳輸的使用者

資料量 (以MB 為單位)，

例如 "^serverDownloadTest" 或

"^serverUploadTest"。

它必須是介於 "4" 和 "512" 之間的值。

(預設值：16)

--^networkiterations 表示將執行的

"^serverDownloadTest" 及/或 "^serverUploadTest"

的反覆項目數目。

它必須是介於 "1" 和 "100" 之間的值。

(預設值：1)

--^diskdatasize 表示將在

"^diskTest" 上寫入然後讀取的資料量。

它必須是介於 "100" 和 "4096" 之間的值。

(預設值：512)

--^disktestpath "^diskTest" 寫入測試檔案的

路徑。如果未提供此參數，

此命令將嘗試使用系統的暫存

路徑。

--^systemdisplaytime 顯示系統資源使用狀況的時間間隔

(以秒為單位)。此選項可供以下

測試使用: "^systemNetworkUsage"

和 "^systemDiskUsage".

它必須是介於 "1" 和 "3600" 之間的值。

(預設: 5 秒)。

--^systemdisplaytimeinterval 系統效能取樣之間的時間間隔

(以秒為單位)。此選項可供

以下測試使用:

"^systemNetworkUsage" 和 "^systemDiskUsage".

它必須是介於 "1" 和 "60" 之間的值。

(預設: 1 秒)。

== CMD\_HELP\_IOSTATS ==

備註:

此命令要求在網路速度測試 ("^serverUploadTest"

及/或 "^serverDownloadTest") 中，使用可用的伺服器。

'--^diskTestPath' 必須指向屬於將要測試的實體

磁碟機。如果未指定任何路徑，此命令會嘗試

使用系統預設的暫存路徑。

指定路徑的磁碟機必須有足夠的空間，才能執行

測試。

命令執行期間，系統會因為執行的測試

而導致效能下降。

可用的測試:

--^serveruploadtest (預設) 測量從

Plastic SCM 用戶端到伺服器的資料上傳速度。

--^serverdownloadtest (預設) 測量從

Plastic SCM 伺服器到用戶端的資料下載速度。

--^disktest (預設) 測量磁碟讀取速度與磁碟

寫入速度。

--^systemnetworkusage 顯示系統網路資源的目前

使用量。

(其顯示 Microsoft Windows 提供的網路介面效能計數器

)。

僅在 Microsoft Windows 中可用。

--^systemdiskusage 顯示系統實體磁碟的目前

使用量。

(其顯示 Microsoft Windows 提供的網路介面

效能計數器)。

僅在 Microsoft Windows 中可用。

範例：

cm ^iostats MYSERVER:8087 --^serveruploadtest --^serverdownloadtest --^nettotalmb=32

== CMD\_DESCRIPTION\_ISSUETRACKER ==

在指定的議題追蹤器中取得、更新或尋找問題狀態。

== CMD\_USAGE\_ISSUETRACKER ==

用法：

cm ^issuetracker <名稱> ^status ^get <任務\_id> <參數>[ ...]

cm ^issuetracker <名稱> ^status ^update <任務\_id> <狀態> <參數>[ ...]

cm ^issuetracker <名稱> ^status ^find <狀態> <參數>[ ...]

cm ^issuetracker <名稱> ^connection ^check <參數>[ ...]

名稱 要連線的議題追蹤器名稱。

目前僅支援 Jira。

任務\_id 要查詢或更新的議題編號。

^status 議題追蹤器中的問題有效狀態。

Jira 參數 (皆為強制):

--^user=<使用者> 要驗證的使用者。

--^password=<密碼> 要驗證的密碼。

--^host=<url> 議題追蹤器的目標 URL。

--^projectkey=<金鑰> Jira 專案的專案金鑰。

== CMD\_HELP\_ISSUETRACKER ==

範例：

cm ^issuetracker jira ^status ^get 11 --^user=user@mail.es --^password=pwd \

--^host=https://user.atlassian.net --^projectkey=PRJ

(取得 'PRJ' 專案的議題 11 狀態。)

cm ^issuetracker jira ^status ^update 11 "完成" --^user=user@mail.es \

--^password=pwd --^host=https://user.atlassian.net --^projectkey=PRJ

(將 'PRJ' 專案的議題 11 狀態更新為 '完成'。)

cm ^issuetracker jira ^status ^find "完成" --^user=user@mail.es --^password=pwd \

--^host=https://user.atlassian.net --^projectkey=PRJ

(取得 'PRJ' 專案狀態已設為 '完成' 的任務 ID -)

cm ^issuetracker jira ^connection ^check --^user=user@mail.es --^password=pwd \

--^host=https://user.atlassian.net --^projectkey=PRJ

(檢查設定參數是否有效。)

== CMD\_DESCRIPTION\_LICENSEINFO ==

顯示授權資訊與授權使用狀況。

== CMD\_USAGE\_LICENSEINFO ==

用法：

cm ^licenseinfo | ^li [--^server=<儲存庫伺服器規格>] [--^inactive] [--^active]

[--^sort=(^name|^status)]

選項:

--^server 從指定的伺服器取得授權資訊。

如果未指定任何伺服器，請在用戶端上設定的

伺服器上執行命令。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫伺服器

規格。)

--^inactive 僅顯示 [授權使用狀況] 區段中的非作用中使用者。

--^active 僅顯示 [授權使用狀況] 區段中的作用中使用者。

--^sort 依其中一個指定的排序選項來排序使用者:

'^name' 或 '^status'。

== CMD\_HELP\_LICENSEINFO ==

備註:

顯示的資源包含截止日期、已啟用和已停用

使用者等。

範例：

cm ^licenseinfo

cm ^licenseinfo --^server=myserver:8084

cm ^licenseinfo --^sort=^name

== CMD\_DESCRIPTION\_LINKTASK ==

將變更集連結至任務。

== CMD\_USAGE\_LINKTASK ==

這是自動化命令，目的僅限用於自動化 'cm'。

方便使用的程度並不高。

用法：

cm ^linktask | ^lt <變更集規格> <副檔名\_前置詞> <任務\_名稱>

變更集規格 連結至任務的完整變更集規格。

(使用 'cm ^help ^objectspec' 以深入瞭解變更集

規格。)

副檔名\_前置詞 要搭配使用之已設定議題追蹤系統

的副檔名前置詞。

任務\_名稱 議題追蹤系統的任務識別碼。

== CMD\_HELP\_LINKTASK ==

範例：

cm ^lt ^cs:8@^rep:default@^repserver:localhost:8084 jira PRJ-1

== CMD\_DESCRIPTION\_LOCK\_LIST ==

顯示伺服器上的鎖定。

== CMD\_USAGE\_LOCK\_LIST ==

用法：

cm ^lock ^list | ^ls [<修訂規格> [ ...]] [--^server=<伺服器>]

[--^onlycurrentuser] [--^onlycurrentworkspace]

[--^ignorecase]

修訂規格 如果存在一或多個，當其相關項目在伺服器中

鎖定時，此命令將針對每個指定的修訂顯示

一個鎖定行。否則，

此命令將列出預設伺服器 (或使用

'--^server' 選項設定的伺服器) 中所有鎖定的項目

使用

多個規格時，請使用空格來分隔修訂規格。

(使用 'cm ^help ^objectspec' 以深入瞭解修訂規格。)

選項:

--^server 儲存庫伺服器規格。

此選項將覆寫從目前工作區或

client.conf 檔案擷取的預設

伺服器。

(使用 'cm ^help ^objectspec' 以深入瞭解

伺服器規格。)

--^onlycurrentuser 篩選結果，並只顯示由目前使用者

執行的鎖定。

--^onlycurrentworkspace 篩選結果，並只顯示在目前工作區上

執行的鎖定 (依名稱比對)。

--^ignorecase 當使用伺服器路徑規格時，忽略路徑的

大小寫區分。搭配這個標誌時，此命令將適用於

"/src/foo.c"，即使該使用者寫入的是 "/sRc/fOO.c"。

== CMD\_HELP\_LOCK\_LIST ==

備註：

此命令將顯示目前在預設伺服器中遭鎖定的項目

清單。此命令也可接受修訂規格的清單；在這種情況下，

將只會顯示屬於所選項目的鎖定。

'--^server=<伺服器>' 可以用來設定將要查詢的預設伺服器。

此命令會顯示一行包含指定伺服器中的各個鎖定：

- 已鎖定項目的 GUID。

- 執行此鎖定的使用者名稱。

- 其中執行鎖定的工作區名稱。

- 已鎖定項目的路徑 (伺服器路徑格式)。

範例：

cm ^lock ^list

cm ^lock ^list --^server=myserver:8084

cm ^lock ^ls ^serverpath:/src/foo.c#^cs:99@default@localhost:8084

cm ^lock ^list ^revid:3521@default ^itemid:2381@secondary --^onlycurrentuser

cm ^lock ^ls --^onlycurrentuser

cm ^lock ^ls --^onlycurrentuser --^onlycurrentworkspace

== CMD\_DESCRIPTION\_LISTUSERS ==

列出使用者和群組。

== CMD\_USAGE\_LISTUSERS ==

用法：

cm ^listusers | ^lu <儲存庫伺服器規格> [--^onlyusers] [--^onlygroups]

[--^filter= <字串\_篩選>]

儲存庫伺服器規格 儲存庫伺服器規格。

(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

選項：

--^onlyusers 只列出使用者。

--^onlygroups 只列出群組。

--^filter 只列出符合所指定篩選的使用者及/或

群組。

== CMD\_HELP\_LISTUSERS ==

範例：

cm ^lu localhost:8084

(列出伺服器中的所有使用者。)

cm ^listusers localhost:8084 --^onlyusers --^filter=m

(只列出名稱含 "m" 伺服器中的使用者。)

== CMD\_DESCRIPTION\_LOCATION ==

傳回名稱含 'cm' 的路徑。

== CMD\_USAGE\_LOCATION ==

用法：

cm ^location

== CMD\_HELP\_LOCATION ==

== CMD\_DESCRIPTION\_LOCK ==

這個命令可讓使用者管理鎖定。

== CMD\_USAGE\_LOCK ==

用法：

cm ^lock <命令> [選項]

命令：

^list | ^ls

^unlock

若要取得每個命令的詳細資訊，請執行：

cm ^lock <命令> --^usage

cm ^lock <命令> --^help

== CMD\_HELP\_LOCK ==

範例：

cm ^lock ^list

cm ^lock

(如果沒有引數，則可選用 '^list'。)

cm ^lock ^ls ^serverpath:/src/foo.c#^cs:99@default@localhost:8084

cm ^lock ^unlock 91961b14-3dfe-4062-8c4c-f33a81d201f5

== CMD\_DESCRIPTION\_LOG ==

取得變更集中修訂的相關資訊。

== CMD\_USAGE\_LOG ==

用法：

cm ^log [<變更集規格> | <儲存庫規格>] [--^from=<變更集規格\_來源>] [--^allbranches]

[--^ancestors] [--^csformat=<字串\_格式>] [--^itemformat=<字串\_格式>]

[--^xml[=<輸出\_檔案>]] [--^encoding=<名稱>]

[--^repositorypaths | --^fullpaths | --^fp]

選項：

變更集規格 此命令將傳回提供規格之變更集內所做的

每項變更。

(使用 'cm ^help ^objectspec' 以深入瞭解變更集

規格。)

儲存庫規格 此命令將列出所指定儲存庫中所做的

所有變更。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫

規格。)

--^from 列出從變更集規格 [變更集規格\_來源] 到變更集規格

[變更集規格] 中每個變更集內所做的

所有變更。

此 [變更集規格\_來源] 變更集不包含在

輸出內。

當已提供儲存庫規格時即忽略。

--^allbranches 顯示變更集建立時所在之全部分支

依據指定間隔所建立變更集

的相關資訊。

--^ancestors 顯示可透過所指定變更集 ([變更集規格]) 的

上層連結與合併連結連接之變更集

相關資訊。如果也提供起始變更集

([變更集規格\_來源])，此變更集將會做為

所有路徑的下限。備註：當使用

此選項時，將不會顯示變更集的變更。

--^csformat 以指定的格式擷取變更集資訊。如需

詳細資訊，請參閱備註。

--^itemformat 以指定的格式擷取項目資訊。如需

詳細資訊，請參閱備註。

--^xml 將 XML 格式的輸出列印至標準輸出。

可以指定輸出檔案。

--^encoding 與 '--^xml' 選項搭配使用，指定要在

XML 輸出中使用的編碼，亦即：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

--^fullpaths, --^fp 如果可能，會強制列印檔案和目錄的

完整工作區路徑。

--^repositorypaths 列印儲存庫路徑 (伺服器路徑)，而非

工作區路徑。(此選項會覆寫

'--^fullpaths' 選項)。

== CMD\_HELP\_LOG ==

備註：

- 如果未指定 '變更集規格' 也未指定任何選項，該命令會顯示

每個分支於最近一個月內所建立個別變更集的

相關資訊。

- 如果只包含選項 '--^from'，該命令會顯示在變更集建立

所在的分支中，從所指定變更集變更至最近變更集之

每個變更集的相關資訊。

- 如果選項 '--^allbranches' 不含間隔，該命令會擷取

如同僅指定 '變更集規格' 時的

相同資訊。

- 如果使用 '--^from'，其輸出會包含'變更集規格\_來源'+1

及依此類推的資訊。

- 用於顯示變更集資訊的儲存庫，即在執行此命令的路徑中

所載入的儲存庫。

這個命令會接受項目的格式字串 ('--^itemformat') 及

變更集的格式字串 ('--^csformat')。

'--^csformat' 的輸出參數如下所示：

{^tab} 插入定位字元空格。

{^newline} 插入新行。

{^changesetid} 變更集號碼。

{^branch} 變更集建立時的所在分支。

{^date} 變更集的日期。

{^owner} 變更集的擁有者。

{^comment} 變更集的註解。

{^items} 涉及此變更集的項目。

{^repository} 其中存在此變更集的儲存庫。

{^repserver} 伺服器名稱。

'--^itemformat' 的輸出參數如下所示：

{^tab} 插入定位字元空格。

{^newline} 插入新行。

{^path} 項目路徑。

{^branch} 變更集建立時的所在分支。

{^date} 變更集的日期。

{^owner} 變更集的擁有者。

{^shortstatus} 列印短格式。請參閱如下。

{^fullstatus} 列印長格式。請參閱如下。

短格式及對應的長格式：

'^A' ^Added

'^D' ^Deleted

'^M' ^Moved

'^C' ^Changed

這些是有效的輸出字串：

--^csformat="{^newline}在 {^date}建立的變更集 {^changesetid}；{^tab} 已變更項目: {^items}。"

--^itemformat="{^newline}項目 {^path} 已在分支 {^branch} 中變更。"

範例：

cm ^log

(顯示每個分支在最近一個月內所建立每個變更集的

相關資訊。)

cm ^log ^cs:16

(顯示變更集建立所在分支中之變更集 16 的

相關資訊。)

cm ^log ^cs:16 --^csformat="{^newline}變更集 {^changesetid}已建立於 \

{^date}；{^tab} 已變更項目: {^items}。"

(以所指定格式，顯示資訊。)

cm ^log --^from=^cs:20 ^cs:50

(顯示從變更集 21 到變更集 50 中，每個變更集內含

個別修訂的相關資訊。)

cm ^log --^from=^cs:20 ^cs:50 --^allbranches

(顯示儲存庫之個別分支的變更集 21 到變更集 50 中，

每個變更集內含個別修訂的

相關資訊。)

cm ^log ^rep:myrep@localhost:8084

(顯示指定儲存庫中所做變更的相關資訊。

執行此命令時不需要任何工作區。)

cm ^log --^from=^cs:20@^rep:mainRep@localhost:8084

(顯示變更集 21 中每個變更集所內含之個別修訂的

相關資訊。執行此命令時不需要任何工作區，因為

已指定了完整的變更集規格。)

== CMD\_DESCRIPTION\_LS ==

列出樹狀目錄的內容。

== CMD\_USAGE\_LS ==

用法：

cm ^ls | ^dir [<路徑>[ ...]] [--^format=<字串\_格式>] [--^symlink]

[--^selector[=<選擇器\_格式>]] [--^tree=<物件\_規格>]

[-^R | -^r | --^recursive]

[--^xml[=<輸出\_檔案>]] [--^encoding=<名稱>]

選項：

路徑 列出要顯示的路徑。使用空格分隔

路徑。

使用雙引號 (" ") 指定含空格的

路徑。

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

--^symlink 將作業套用至符號連結，而不是套用至

目標。

--^selector 取得此作用中工作區選擇器中的內容。

如已指定 [選擇器\_格式]，則會列出

指定的選擇器。

這個選擇器幾乎已棄用，因為這些選擇器

自 4.x 版起已不再是 Plastic SCM 的中心部分。

--^tree 在指定的變更集或分支中，列出樹狀目錄。

(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

-^R 遞迴列出。

--^xml 將 XML 格式的輸出列印至標準輸出。

可以指定輸出檔案。

--^encoding 與 '--^xml' 選項搭配使用，指定要在

XML 輸出中使用的編碼，亦即：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

== CMD\_HELP\_LS ==

備註：

- 路徑輸入可使用中繼字元。

- 實際清單依工作區選擇器而定。

- 此命令的輸出可依指定的格式字串格式化處理。

- 如果指定了--^tree' 或 --^selector' 選項，則指定的

路徑必須是伺服器路徑 (又稱為 'cm path'): /dir/file.txt，

而非工作區路徑: C:\Users\myuser\mywk\dir\file.txt

- 如未提供任何路徑，則工作區路徑應該是目前的

目錄。如果使用了 '--^tree' 或 '--^selector' 選項，則

應該是使用根目錄路徑 ("/")。

預設格式字串：

"{^size,10} {^date:dd/MM/yyyy} {^date:HH:mm}\

{^type,-6} {^location,-12} {^checkout,-5} {^name}\

{^symlinktarget}"

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{^size}

{^formattedsize}

{^date}

{^type}

^dir 目錄；

^txt 文字檔；

^File 檔案。

{^location} 範例：^br:branch#cset

{^checkout}

{^name}

{^changeset}

{^path}

{^repspec}

{^owner}

{^revid}

{^parentrevid}

{^itemid}

{^brid}

{^repid}

{^server}

{^symlinktarget}

{^hash}

{^chmod}

{^wkpath} 工作區根目錄的相對路徑

{^branch}

{^newlocation} cset@branch

{^guid} (解析所花時間較長)

{^itemguid}

{^transformed} 顯示所轉換項目上的套用規則

您可以自訂 PLASTIC\_LS\_FORMAT 環境變數

'^ls' 格式設定。

範例：

cm ^ls

cm ^ls c:\workspace\src

cm ^ls --^format={^name}

(僅檔名。)

cm ^ls --^symlink

(顯示符號連結的相關資訊，而不顯示「已加上符號」的檔案或目錄的

相關資訊。可用於 UNIX 環境。)

cm ^ls code --^selector

(顯示目前工作區選擇器中 'code' 子目錄的

內容。)

cm ^ls /code --^selector="^rep 'myrep' ^path '/' ^branch '/^main'"

(顯示所指定選擇器上子目錄 '/code' 的內容。

請注意，此路徑是以伺服器格式指定。)

cm ^ls /code --^tree=44@myrep@denver:7070

(列出位於伺服器 'denver:7070' 儲存庫 'myrep' 變更集 44 的

子目錄 '/code'。)

cm ^ls /code --^tree=^br:/main/scm13596@myrep@denver:7070

(列出位在伺服器 'denver:7070' 儲存庫 'myrep' 分支

'/main/scm13596' 中最新變更集的子目錄 '/code'。)

cm ^ls /code --^tree=ae1390ed-7ce9-4ec3-a155-e5a61de0dc77@myrep@denver:7070

(列出位於伺服器 'denver:7070' 儲存庫 'myrep'

變更集 ae1390ed-7ce9-4ec3-a155-e5a61de0dc77 的

子目錄 '/code'。)

== CMD\_DESCRIPTION\_TRIGGER\_LIST ==

在伺服器上，列出指定類型的觸發器。

== CMD\_USAGE\_TRIGGER\_LIST ==

用法：

cm ^trigger | ^tr ^list | ^ls [<子類型-類型>] [--^server=<儲存庫伺服器規格>]

[--^format=<字串\_格式>]

選項：

子類型-類型 觸發器執行與觸發器作業。

輸入 'cm ^showtriggertypes' 以查看觸發器

類型的清單。

--^server 在指定的伺服器上列出觸發器。

如果未指定任何伺服器，請在用戶端上設定的

伺服器上執行命令。

(使用 'cm ^help ^objectspec' 以深入瞭解伺服器

規格。)

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

== CMD\_HELP\_TRIGGER\_LIST ==

備註：

如果未指定此類型，將列出伺服器上的所有觸發器。

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{0} 觸發器位置。

{1} 觸發器名稱。

{2} 觸發器名稱。

{3} 觸發器名稱。

{4} 觸發器類型。

{5} 觸發器篩選。

範例：

cm ^trigger list after-mklabel

cm ^tr ^ls ^before-mkbranch --^server=myserver:8084

== CMD\_DESCRIPTION\_MANIPULATESELECTOR ==

將選擇器變更為時間。

== CMD\_USAGE\_MANIPULATESELECTOR ==

這是自動化命令，目的僅限用於自動化 'cm'。

方便使用的程度並不高。

用法：

cm ^manipulateselector | ^ms [<工作區\_路徑> | <工作區\_規格>] --^atdate=<選擇器\_日期>

工作區\_路徑 工作區的路徑。

工作區\_規格 工作區規格。(使用 'cm ^help ^objectspec'

以深入瞭解規格。)

選項：

--^atdate 傳回將重建工作區的選擇器，

因其已搜尋到指定的日期。

== CMD\_HELP\_MANIPULATESELECTOR ==

備註：

如果未指定路徑或工作區規格，命令將以

目前目錄做為工作區路徑。

範例：

cm ^manipulateselector c:\workspace --^atdate=yyyy-MM-ddTHH:mm:ss

cm ^manipulateselector --^atdate=yyyy-MM-ddTHH:mm:ss

cm ^manipulateselector > mySelector.txt --^atdate=yyyy-MM-ddTHH:mm:ss

cm ^manipulateselector ^wk:build\_wk@BUILDER --^atdate=yyyy-MM-ddTHH:mm:ss

== CMD\_DESCRIPTION\_MERGE ==

合併分支與另一個分支。

== CMD\_USAGE\_MERGE ==

用法：

cm ^merge <來源\_規格> [--^merge] [--^cherrypicking] [--^forced]

[--^mergetype=(^onlyone|^onlysrc|^onlydst|^try|^forced)]

[--^interval-origin=<變更集規格> | --^ancestor=<變更集規格>]

[--^keepsource | --^ks] [--^keepdestination | --^kd]

[--^automaticresolution=<衝突-類型>[;...]]

[--^subtractive] [--^mount] [--^printcontributors]

[--^noprintoperations] [--^silent]

[(--^to=<分支規格> | --^destination=<分支規格>)[--^shelve]]

[--^no-dst-changes]

[-^c=<字串\_註解> | --^commentsfile=<註解\_檔案>]

[--^resolveconflict --^conflict=<索引>

--^resolutionoption=(^src|^dst|(^rename --^resolutioninfo=<字串名稱>))

--^mergeresultfile=<路徑> --^solvedconflictsfile=<路徑>]

[--^nointeractiveresolution]

[--^machinereadable [--^startlineseparator=<分隔符號>]

[--^endlineseparator=<分隔符號>] [--^fieldseparator=<分隔符號>]]

來源\_規格 要從中合併之來源物件的規格：

- 分支規格：'[^br:/]br\_name'

- 標籤規格：'^lb:lb\_name'

- 變更集規格：'^cs:cs\_number'

- 擱置規格：'^sh:shelve\_number'

(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

選項：

--^merge 執行合併。否則，列印找到的

衝突。

--^cherrypicking 合併包含在來源變更集內的

變更。如果合併來源規格是個標籤，

則不使用這個選項。

--^forced 如果來源與目的地變更集已經連接，

則不用檢查。

這個選項只能用於間隔合併

和揀選。

--^mergetype 如需詳細資訊，請參閱備註。

--^interval-origin 指定該選擇哪個變更集來做為

間隔起點，使合併僅處理

在來源變更集與所指定

間隔起點之間的不同內容。

--^ancestor 這是 --^interval-origin. 的別名。

--^keepsource 接受來源貢獻者中，項目有衝突的

所有變更。

--^keepdestination 保留目的地貢獻者中，項目有衝突的

變更。

--^automaticresolution 用於解決目錄衝突。這個選項

可讓您選擇應自動選取

來源貢獻者或目的地貢獻者

來解決衝突。

使用分號來分隔衝突類型。

如需詳細資訊，請參閱備註。

--^subtractive 刪除因合併所引入的變更。傳遞至

命令的參數 (來源\_規格)

會用於指定要刪除變更的

來源位置。此來源必定是變更集。如果是

變更集間隔，則必須使用 '--^interval-origin'

來定義間隔起點。為了移除

變更，系統會建立全新的已簽出修訂，

其中將包含上一個項目的內容，

只是不包含已刪除的變更。

--^mount 指定儲存庫的裝載點。

--^printcontributors 列印出貢獻者 (基底、來源與

目的地)。

--^noprintoperations 以無訊息方式解決合併，完全

不顯示關於解決方法的資訊。

--^silent 不顯示任何輸出。

--^to | --^destination 運用完整衝突解決方法，

將合併目標作業執行至所指定的分支

(透過輸入分支規格或 brspec)。

「合併目標」(或無工作區合併) 是在伺服器端

完成的合併。而一般合併

是「來源」分支、標籤或變更集進行的工作區合併，

整個合併目標作業會完全

在伺服器上完成。在一般合併中，

「目的地」是指工作區，而在「合併目標」中，

一定要指定目的地 (這樣才合乎其名稱中的

「目標」)。

如需詳細資訊，請查看以下關於

「合併目標」功能的連結：

https://www.plasticscm.com/download/help/mergeto

--^shelve 建立包含合併結果 (加上合併追蹤資訊)

的擱置，而不是建立

新的變更集。當合併來源為擱置時，

這個選項將不可使用。這個

選項僅能用於 server-side-merge

(又名為「合併目標」)。因此，必須指定 '--^to' 和

'--^merge' 選項。

--^no-dst-changes 確定目的地貢獻者沒有任何變更

(目的地變更集也是共同

上層)。當目的地發生變更時，

將不允許進行合併。

-^c 將指定的註解套用至合併作業中

建立的變更集。

--^commentsfile 將指定檔案中的註解套用至合併作業中

建立的變更集。

--^resolveconflict (主要由外掛程式所使用。如需詳細資訊，請參閱備註。)

用於解決目錄衝突。

--^conflict 搭配 '--^resolveconflict' 標誌使用，指定

要解決衝突的索引，從 1 開始。

--^resolutionoption 搭配 '--^resolveconflict' 標誌使用，指示

解決衝突的類型。使用下列

其中一個選項：'^src'、'^dst'、'^rename'。

如需詳細資訊，請參閱備註。

--^resolutioninfo 搭配 '--^resolveconflict' 標誌使用，提供

當 '--^resolutionoption' 選項為 'rename' 時

將要使用的名稱。

--^mergeresultfile 搭配 '--^resolveconflict' 標誌使用，會將

不同呼叫之間所造成的衝突資訊

輸出至檔案。指定的路徑

將於第一個呼叫期間建立，並在後續的個別呼叫中

逐一更新。

--^solvedconflictsfile 搭配 '--^resolveconflict' 標誌使用，將

不同呼叫之間所解決衝突的資訊

輸出至檔案。指定的

路徑將在第一個呼叫期間建立，並在後續的個別呼叫中

逐一更新。

--^nointeractiveresolution (主要由外掛程式所使用。如需詳細資訊，請參閱備註。)

會避免提示使用者進行手動衝突解決。

如此將不解決目錄衝突。

--^machinereadable (主要由外掛程式所使用。如需詳細資訊，請參閱備註。)

以易於剖析的格式輸出結果。

--^startlineseparator 搭配 '--^machinereadable' 標誌使用，指定

各行應如何起始。(預設：空的字串。)

--^endlineseparator 搭配 '--^machinereadable' 標誌使用，指定

各行應如何結束。(預設：空的字串。)

--^fieldseparator 搭配 '--^machinereadable' 標誌使用，指定

應如何分隔欄位。(預設：

空格。)

== CMD\_HELP\_MERGE ==

備註：

這個命令可用來合併兩個分支之間的變更，或是標籤與分支

之間的變更。合併的目的地一定是個分支。

合併來源會以引數方式指定。

目的地是工作區的目前內容。

例如，若要顯示將從分支 task001 合併至主要分支

的元素，選擇器必須指向主要分支，工作區

一定會更新，並接著：

cm ^merge ^br:/task001

若要確實實際此合併，則要加上 '--^merge' 選項：

cm ^merge ^br:/task001 --^merge

若要定義合併來源，可以使用下列規格：

- 分支規格 (brspec)：

[^br:/]br\_name

範例：^br:/main/task001

(上面範例是執行從此分支中的最新變更集所進行的合併作業。)

- 標籤規格 (lbspec)：

^lb:lb\_name

範例：^lb:BL001

(從標記的變更集進行合併。)

- 變更集規格：

^cs:cs\_number

範例：^cs:25

(從指定的變更集內容進行合併。)

- 擱置規格 (shspec)：

^sh:shelve\_number

範例：^sh:2

(從指定的擱置內容進行合併。)

若要自動解決目錄衝突，請使用 '--^automaticresolution' 選項，

並在指定衝突類型後指定必須在合併作業期間

選取的貢獻者 (來源或目的地)。

(使用分號 (;) 來分隔個別的「衝突類型-貢獻者」配對。)

例如:

cm ^merge ^cs:2634 --^merge --^automaticresolution=^eviltwin-src;^changedelete-src

(從變更集 2634 進行的合併作業，可以解決 "^eviltwin" 與

"^changedelete" 衝突，因這個作業會在兩邊都保留來源 ("-^src")

貢獻者。)

- 接在衝突類型之後的 "-^src" 後綴，將會指示合併命令保留

來源貢獻者變更。

- 如果是 "-^dst" 後綴，則將保留目的地貢獻者變更。

下面是合併命令可支援的衝突類型清單：

"^movedeviltwin"、"^eviltwin"、"^changedelete"、"^deletechange"、"^movedelete"、

"^deletemove"、"^loadedtwice"、"^addmove"、"^moveadd"、"^divergentmove",

"^cyclemove"、"^all".

"^all" 值會覆寫其他的選項。在下面範例中，

將忽略 "^eviltwin-dst"。

cm ^merge ^br:/main/task062 --^merge --^automaticresolution=^all-src;^eviltwin-dst

請查看下面連結，以深入瞭解合併衝突：

https://www.plasticscm.com/download/help/directorymerges

下面是 '--^mergetype' 的可用選項：

^onlyone 在只有一個貢獻者修改過項目時

自動合併。

^onlysrc 在只有來源貢獻者修改過項目時

自動合併。

^onlydst 在只有目的地貢獻者修改過項目時

自動合併。

^try 在只有一個貢獻者修改過程式碼的衝突片段

(個別衝突) 時自動合併。

^forced 一律嘗試解決所有非自動衝突。

下面列出主要由外掛程式與整合作業所使用的選項：

- '--^resolveconflict' 可解決目錄衝突。這時也必須使用

下列選項：

- '--^conflict' 是所要解決衝突的索引，

從 1 開始。

- '--^resolutionoption' 指示要使用的衝突解決

方法。可選取的解決方法如下：

- '^src' 可保留來源變更，並捨棄

目的地變更

- '^dst' 可保留目的地變更，並捨棄

來源變更

- '^rename' (僅在衝突類型支援這個解決方法時

適用) 可將目的地重新命名成搭配

'--^resolutioninfo' 選項所提供的指定名稱。

- '--^resolutioninfo' 可提供將用於 '^rename' 解決方法

的名稱

- '--^mergeresultfile' 與 '--^solvedconflictsfile'，都可用於

儲存不同呼叫之間的合併資訊。

- '--^nointeractiveresolution' 可指示合併不要求使用者進行

手動衝突解決方法。

- '--^machinereadable' 和 '--^startlineseparator'、'--^endlineseparator'、

'--^fieldseparator' 等選項，可將輸出列印成電腦可讀取的

格式 (易於剖析的)。

範例：

cm ^merge --^machinereadable --^startlineseparator=start@\_@line \

--^endlineseparator=new@\_@line --^fieldseparator=def#\_#sep \

--^mergeresultfile=C:\Users\Borja\AppData\Local\Temp\2tmp4D6C.tmp \

--^solvedconflictsfile=C:\Users\Borja\AppData\Local\Temp\2tmp4D6D.tmp \

--^resolveconflict --^conflict=1 --^resolutionoption=rename \

--^resolutioninfo=目錄\_目的地 ^br:/main/task --^merge

範例：

cm ^merge ^br:/task001

(不合併，只列印出要合併的項目。)

cm ^merge ^br:/task001 --^merge

(執行從分支 'task001' 進行的合併。)

cm ^merge ^cs:5 --^merge --^cherrypicking --^interval-origin=^cs:2

(從變更集間隔 [2,5] 揀選。)

cm ^merge ^cs:8 --^merge --^subtractive --^keepdestination

(從變更集 8 削減合併，保留目的地中有衝突的

元素變更。)

cm ^merge ^br:/main/task001 --^to=^br:/main --^merge -^c="整合式新 UI"

(執行伺服器端合併，又名「合併目標」，將從分支 'task001' 合併至

分支 'main'，並設定註解。)

cm ^merge ^br:/main/task001 --^to=^br:/main --^merge --^shelve

(執行從分支 'task001' 進行伺服器端合併至分支 'main'，並在擱置上

保留結果。)

cm ^merge ^sh:2 --^to=^br:/main --^merge --^no-dst-changes

(只有在結果是從目前 'main' 前端所建立時，才將擱置 2 套用

至 'main'。)

== CMD\_DESCRIPTION\_ATTRIBUTE\_CREATE ==

建立新屬性。

== CMD\_USAGE\_ATTRIBUTE\_CREATE ==

用法：

cm ^attribute | ^att ^create | ^mk <屬性\_名稱>

屬性\_名稱 屬性名稱

== CMD\_HELP\_ATTRIBUTE\_CREATE ==

範例：

cm ^attribute ^create 狀態

(建立屬性 '狀態'。)

cm ^att ^mk 整合

(建立屬性 '整合'。)

== CMD\_DESCRIPTION\_BRANCH ==

可讓使用者管理分支。

== CMD\_USAGE\_BRANCH ==

用法：

cm ^branch | ^br <命令> [選項]

命令：

^create | ^mk

^delete | ^rm

^rename

^history

^showmain

^showmerges

若要取得每個命令的詳細資訊，請執行：

cm ^branch <命令> --^usage

cm ^branch <命令> --^help

== CMD\_HELP\_BRANCH ==

範例：

cm ^branch /main/scm21345

cm ^branch ^create /main/scm21345

cm ^branch ^delete /main/scm21345

cm ^branch ^rename /main/scm21345 scm21346

cm ^branch ^history /main/scm21345

cm ^branch ^showmain

cm ^branch ^showmerges file.txt

== CMD\_DESCRIPTION\_BRANCH\_CREATE ==

建立新分支。

== CMD\_USAGE\_BRANCH\_CREATE ==

用法：

cm ^branch | ^br [^create | ^mk] <分支規格>

[--^changeset=<變更集規格> | --^label=<標籤規格>]

[-^c=<字串\_註解> | -^commentsfile=<註解\_檔案>]

分支規格 新分支的名稱或規格。

(使用 'cm ^help ^objectspec' 以深入瞭解分支規格。)

選項：

--^changeset 用作新分支起點的變更集。

(使用 'cm ^help ^objectspec' 以深入瞭解變更集規格。)

--^label 用作新分支起點的標籤。

(使用 'cm ^help ^objectspec' 以深入瞭解標籤規格。)

-^c 將指定的文字填入新分支的

註解欄位。

-^commentsfile 將所指定檔案的內容填入

新分支的註解欄位。

== CMD\_HELP\_BRANCH\_CREATE ==

備註：

若要建立最上層分支，請指定不含任何階層的名稱。

例如:

cm ^br /dev

如果未指定選擇性參數 '--^changeset'，新分支的基底將是

上層分支的最後一個變更集。如果新分支

是最上層分支，則使用的基底變更集將會是 cset 0。

您可以使用 '-^c' 或 '-^m' 切換來指定註解：

cm ^branch /main/task001 -^c="這是註解"

cm ^branch /main/task001 -^m "這是註解"

設定 PLASTICEDITOR 環境變數，以指定要用來輸入註解

的編輯器。如果已經設定 PLASTICEDITOR 環境變數，

且註解是空的，編輯器將自動啟動，並允許您

指定註解。

範例：

cm ^branch task001

cm ^branch ^create task001

cm ^branch ^mk task001

cm ^br ^mk task001

(在目前工作區的儲存庫中，建立最上層的分支

'task001'。)

cm ^branch ^br:/task001/task002@

(建立 'task002' 來做為 'task001' 的下層分支。)

cm ^br /main/task001@myrep@myserver:8084 -^c="我的註解"

(建立 'task001' 分支做為儲存庫 'myrep@myserver:8084'

中 'main' 的下層分支，加上 '我的註解'。)

cm ^branch ^br:/main/task001 --^changeset=2837 -^commentsfile=commenttask001.txt

(建立 'task001' 分支做為 'main' 的下層分支，基底為 'changeset=2837'，

並將註解套用至 'commenttask001.txt' 檔案。)

== CMD\_DESCRIPTION\_BRANCH\_DELETE ==

刪除一或多個分支。

== CMD\_USAGE\_BRANCH\_DELETE ==

用法：

cm ^branch | ^br ^delete | ^rm <分支規格>[ ...]

分支規格 要刪除的分支。使用空格分隔分支。

(使用 'cm ^help ^objectspec' 以深入瞭解分支

規格。)

== CMD\_HELP\_BRANCH\_DELETE ==

備註：

此命令刪除一或多個分支。

範例：

cm ^branch ^delete /main/task001

(刪除屬於目前工作區中儲存庫內 'main' 的下層分支，

名稱為 'task001' 的分支。)

cm ^br ^rm main/task002 /main/task012@reptest@myserver:8084

(刪除目前工作區之儲存庫內的分支 '/main/task002'，

以及儲存庫 'reptest@myserver:8084' 內的分支 '/main/task012'。)

== CMD\_DESCRIPTION\_BRANCH\_RENAME ==

重新命名分支。

== CMD\_USAGE\_BRANCH\_RENAME ==

用法：

cm ^branch | ^br ^rename <分支規格> <新\_名稱>

分支規格 要重新命名的分支。

(使用 'cm ^help ^objectspec' 以深入瞭解分支規格。)

新\_名稱 分支的新名稱。

== CMD\_HELP\_BRANCH\_RENAME ==

備註：

此命令會重新命名分支。

範例：

cm ^branch ^rename /main/task0 task1

(將分支 '/main/task0' 重新命名成 '/main/task1'。)

cm ^br ^rename ^br:/main@reptest@server2:8084 secondary

(將儲存庫 'reptest' 的分支 'main' 重新命名成 'secondary'。)

== CMD\_DESCRIPTION\_BRANCH\_HISTORY ==

顯示分支的歷史記錄。

== CMD\_USAGE\_BRANCH\_HISTORY ==

用法：

cm ^branch | ^br ^history <分支規格> [--^dateformat=<日期\_格式>]

[--^machinereadable]

分支規格 要保存其歷史記錄的分支規格。

(使用 'cm ^help ^objectspec' 以深入瞭解分支規格。)

選項：

--^dateformat 用於輸出日期的格式。

--^machinereadable 以易於剖析的格式輸出結果。

== CMD\_HELP\_BRANCH\_HISTORY ==

範例：

cm ^branch ^history ^br:/main/scm001@myrepository@myserver:8084

(顯示 'myserver' 伺服器上，'myrepository' 儲存庫之分支 '/main/scm001'

的歷史記錄。)

cm ^br ^history main --^dateformat="yyyy, dd MMMM" --^machinereadable

(以指定日期格式與易於剖析的格式，顯示目前儲存庫

之 'main' 分支的歷史記錄。)

== CMD\_DESCRIPTION\_BRANCH\_SHOWMAIN ==

顯示儲存庫的主要分支。

這是自動化命令，目的僅限用於自動化 'cm'。

儲存庫的主要分支很有可能是 '/main'。

== CMD\_USAGE\_BRANCH\_SHOWMAIN ==

用法：

cm ^branch | ^br ^showmain [<儲存庫規格>] [--^encoding=<名稱>]

[--^format=<格式\_字串>] [--^dateformat=<日期\_格式>]

儲存庫規格 其中將顯示主要分支的

儲存規格。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫規格。)

選項：

--^encoding 指定要用於輸出的編碼，

例如：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

--^dateformat 用於輸出日期的格式。

== CMD\_HELP\_BRANCH\_SHOWMAIN ==

備註：

這個命令會顯示儲存庫的主要分支。

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{^id} 分支 ID。

{^comment} 註解。

{^date} 日期。

{^name} 名稱。

{^owner} 擁有者。

{^parent} 上層分支名稱。

{^repository} 儲存庫。

{^repname} 儲存庫名稱。

{^repserver} 伺服器。

{^changeset} 分支的前端變更集。

範例：

cm ^branch ^showmain

(顯示目前工作區之儲存庫的主要分支。)

cm ^branch ^showmain repo@server:8084

(顯示伺服器 'server:8084' 內，儲存庫 'repo' 的

主要分支。)

cm ^br ^showmain --^dateformat="yyyy, dd MMMM" --^encoding=utf8

(以指定的日期格式顯示儲存庫的主要分支，

且輸出的編碼為 utf8。)

cm ^br ^showmain --^format="{^id} - {^name}"

(顯示儲存庫的主要分支，且只列印其 ID 與名稱。)

== CMD\_DESCRIPTION\_BRANCH\_SHOWMERGES ==

顯示待合併的分支。

== CMD\_USAGE\_BRANCH\_SHOWMERGES ==

這是自動化命令，目的僅限用於自動化 'cm'。

方便使用的程度並不高。

用法：

cm ^branch | ^br ^showmerges <項目\_路徑>[ ...]

[--^format=<格式\_字串>]

[--^dateformat=<日期\_格式>]

選項：

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

--^dateformat 用於輸出日期的格式。

== CMD\_HELP\_BRANCH\_SHOWMERGES ==

備註：

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{^id} 分支 ID。

{^comment} 註解。

{^date} 日期。

{^name} 名稱。

{^owner} 擁有者。

{^parent} 上層分支名稱。

{^parentid} 上層分支 ID。

{^repid} 儲存庫 ID。

{^repository} 儲存庫。

{^repname} 儲存庫名稱。

{^repserver} 儲存庫伺服器。

範例：

cm ^branch ^showmerges file.txt

(顯示涉及 'file.txt' 待合併作業的分支。)

cm ^branch ^showmerges file.txt --^format="{^date} {^name}" --^dateformat="yyMMdd"

(顯示涉及合併作業的分支，並以指定的日期格式，

只列印日期與名稱。)

== CMD\_DESCRIPTION\_REPOSITORY ==

可讓使用者管理儲存庫。

== CMD\_USAGE\_REPOSITORY ==

用法：

cm ^repository | ^repo <命令> [選項]

命令：

^create | ^mk

^delete | ^rm

^list | ^ls

^rename

^add

若要取得每個命令的詳細資訊，請執行：

cm ^repository <命令> --^usage

cm ^repository <命令> --^help

== CMD\_HELP\_REPOSITORY ==

範例：

cm ^repository

cm ^repository ^list

cm ^repository 新儲存庫

cm ^repository ^create 新儲存庫

cm ^repository ^rename 舊名稱 新名稱

cm ^repository ^add C:\repo\

== CMD\_DESCRIPTION\_REPOSITORY\_CREATE ==

在伺服器上建立儲存庫。

== CMD\_USAGE\_REPOSITORY\_CREATE ==

用法：

cm ^repository | ^repo <儲存庫\_名稱>

cm ^repository | ^repo <儲存庫伺服器規格> <儲存庫\_名稱>[ ...]

cm ^repository | ^repo [^create | ^mk] <儲存庫\_名稱>

儲存庫伺服器規格 儲存庫伺服器規格。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫伺服器

規格。)

儲存庫\_名稱 新儲存庫或多個儲存的名稱或多個名稱。

使用空格分隔儲存庫名稱。

== CMD\_HELP\_REPOSITORY\_CREATE ==

範例：

cm ^repository MyRep

cm ^repo 192.168.1.140:8087 Rep01 Rep01/ModuleA Rep01/ModuleB

cm ^repo ^create Rep01

cm ^repo ^mk 清單

== CMD\_DESCRIPTION\_REPOSITORY\_DELETE ==

從伺服器中刪除儲存庫。

== CMD\_USAGE\_REPOSITORY\_DELETE ==

用法：

cm ^repository | ^repo ^delete | ^rm <儲存庫規格>

選項：

儲存庫規格 儲存庫規格。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫規格。)

== CMD\_HELP\_REPOSITORY\_DELETE ==

備註：

從儲存庫伺服器中刪除儲存庫。

資料並未從資料庫後端中移除，但已拔除連線，

因此無法再存取。

(之後可重新連接資料，請參閱 'cm ^repository ^add'。)

範例：

cm ^repository ^delete myrepository@^repserver:myserver:8084

cm ^repository ^rm myrepository@myserver:8084

cm ^repo ^rm myrepository

== CMD\_DESCRIPTION\_REPOSITORY\_LIST ==

列出伺服器上的儲存庫。

== CMD\_USAGE\_REPOSITORY\_LIST ==

用法：

cm ^repository | ^repo [^list | ^ls] [<儲存庫伺服器規格>] [--^format=<字串\_格式>]

選項：

儲存庫伺服器規格 儲存庫伺服器規格。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫伺服器

規格。)

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

== CMD\_HELP\_REPOSITORY\_LIST ==

備註：

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{^repid} | {0} 儲存庫識別碼。

{^repname} | {1} 儲存庫名稱。

{^repserver} | {2} 伺服器名稱。

{^repowner} | {3} 儲存庫擁有者。

{^repguid} | {4} 儲存庫的唯一識別碼。

{^tab} 插入定位字元空格。

{^newline} 插入新行。

如果格式參數值是 '^TABLE'，輸出將使用

資料表格式與 {^repid}、{^repname} 和 {^repserver} 等欄位印出。

範例：

cm ^repository

(列出所有儲存庫。)

cm ^repository ^list localhost:8084 --^format="{1, -20} {3}"

(在 20 個空格中寫入儲存庫名稱，靠左對齊，接著接上空白，

再寫入儲存庫擁有者。)

cm ^repository ^ls localhost:8084 --^format="{^repname, -20} {^repowner}"

(如先前範例方式，寫入名稱。)

cm ^repo ^ls localhost:8084 --^format=^TABLE

(使用資料格式，以及儲存庫 ID、儲存庫名稱、儲存庫伺服器

名稱欄位，寫入儲存庫的清單。)

== CMD\_DESCRIPTION\_REPOSITORY\_RENAME ==

重新命名儲存庫。

== CMD\_USAGE\_REPOSITORY\_RENAME ==

用法：

cm ^repository | ^repo ^rename [<儲存庫規格>] <新\_名稱>

儲存庫規格 要重新命名的儲存庫。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫

規格。)

新\_名稱 儲存庫的新名稱。

== CMD\_HELP\_REPOSITORY\_RENAME ==

備註：

此命令會重新命名儲存庫。

如果未指定儲存格規格，將直接使用目前儲存庫。

範例：

cm ^repository ^rename development

(目前的儲存庫將重新命名成為 'development'。)

cm ^repo ^rename ^rep:default@SERVER:8084 development

('SERVER' 上的 'default' 儲存庫將重新命名為 'development'。)

== CMD\_DESCRIPTION\_REPOSITORY\_ADD ==

以新增資料庫方式，連接現有的儲存庫。

== CMD\_USAGE\_REPOSITORY\_ADD ==

用法：

cm ^repository | ^repo ^add <資料庫\_檔案> <儲存庫\_名稱> <儲存庫伺服器規格>

資料庫\_檔案 在資料庫後端之資料庫檔案的名稱。

儲存庫\_名稱 儲存庫的名稱。

儲存庫伺服器規格 儲存庫伺服器規格。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫

伺服器規格。)

== CMD\_HELP\_REPOSITORY\_ADD ==

備註：

將現有的儲存庫資料庫重新連接至伺服器。

範例：使用 'cm ^repository ^delete' 命令後，使用 '^add' 命令，

將儲存庫從一部伺服器中移動到其他伺服器，或是重新還原

封存的儲存庫。

範例：

cm ^repository ^add rep\_27 myrepository myserver:8084

== CMD\_DESCRIPTION\_TRIGGER\_CREATE ==

建立在伺服器上的新觸發器。

== CMD\_USAGE\_TRIGGER\_CREATE ==

用法：

cm ^trigger | ^tr ^create | ^mk <子類型-類型> <新\_名稱> <指令碼\_路徑>

[--^position=<新\_位置>]

[--^filter=<字串\_篩選>]

[--^server=<儲存庫伺服器規格>]

子類型-類型 觸發器執行與觸發器作業。

輸入 'cm ^showtriggertypes' 以查看觸發器

類型的清單。

新\_名稱 新觸發器的名稱。

指令碼\_路徑 要執行之指令碼所在的伺服器的

磁碟路徑。如果命令行開頭為 "^webtrigger"，

觸發器將視為 Web 觸發器。如需

如需詳細資訊，請參閱備註。

選項：

--^position 指定觸發器的新位置。

此位置不得由其他相同類型的觸發器

使用。

--^filter 僅檢查符合已指定篩選條件的項目。

--^server 建立在所指定伺服器上的觸發器。

如果未指定任何伺服器，請在用戶端上設定的

伺服器上執行命令。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫

伺服器規格。)

== CMD\_HELP\_TRIGGER\_CREATE ==

備註：

Web 觸發器：藉由輸入 "^webtrigger <目標-uri>"

做為觸發器命令，以建立 Web 觸發器。在此情況下，觸發器將對應指定的 URI 來執行

POST 查詢，其中要求本文包含具有觸發器

環境變數的 JSON 字典，以及指向字串陣列

的 INPUT 金鑰。

範例：

cm ^trigger ^create ^after-setselector "BackupMgr" "/path/to/script" --^position=4

cm ^tr ^mk ^before-mklabel new "/path/to/script" --^server=myserver:8084

cm ^tr ^mk ^after-mklabel Log "/path/to/script" --^filter="^rep:myRep,LB\*"

(只有當標籤名稱以 'LB' 為開頭，且建立所在儲存庫名為

'myRep' 時，這個觸發器才會執行。)

cm ^tr ^mk ^after-checkin NotifyTeam "^webtrigger http://myserver.org/api"

== CMD\_DESCRIPTION\_MOVE ==

移動或重新命名檔案或目錄。

== CMD\_USAGE\_MOVE ==

用法：

cm ^move | ^mv <來源\_路徑> <目的地\_路徑> [--^format=<字串\_格式>]

[--^errorformat=<字串\_格式>]

來源\_路徑 來源項目路徑。

目的地\_路徑 目的地項目路徑。

選項：

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

--^errorformat 以指定的格式擷取錯誤訊息。如需

詳細資訊，請參閱備註。

== CMD\_HELP\_MOVE ==

備註：

這個命令會移動或重新命名儲存庫中的項目。

變更也是在本機檔案系統中完成。

如果來源路徑是檔案，則目的地路徑可能為檔案或

目錄。在第一個情況中，會重新命名檔案；否則，會

移動項目。

如果來源路徑是個目錄，目的地路徑必須是目錄。

要移動或重新命名的項目必須存在。

格式：

{0} 來源路徑 (同時適用於 '--^format' 和 '--^errorformat')

{1} 目的地路徑 (同時適用於 '--^format' 和 '--^errorformat')

範例：

cm ^move file.txt file.old

(重新命名項目。)

cm ^mv .\file.old .\oldFiles

(將 'file.old' 移動到 'oldFiles'。)

cm ^move .\src .\src2

(重新命名目錄。)

== CMD\_DESCRIPTION\_LABEL ==

可讓使用者管理標籤。

== CMD\_USAGE\_LABEL ==

用法：

cm ^label | ^lb <命令> [選項]

命令：

^create | ^mk

^delete | ^rm

^rename

若要取得每個命令的詳細資訊，請執行：

cm ^label <命令> --^usage

cm ^label <命令> --^help

== CMD\_HELP\_LABEL ==

範例：

cm ^label 我的新標籤 ^cs:42

('^create' 命令是選擇性。)

cm ^label ^rename 我的新標籤 新標籤名稱

cm ^label ^delete 新標籤名稱

== CMD\_DESCRIPTION\_LABEL\_CREATE ==

將標籤套用至變更集，並視需要建立標籤。

== CMD\_USAGE\_LABEL\_CREATE ==

用法：

cm ^label [^create] <分支規格> [<變更集規格> | <工作區\_路徑>]

[--^allxlinkedrepositories]

[-^c=<字串\_註解> | -^commentsfile=<註解\_檔案>]

分支規格 新標籤名稱。

(使用 'cm ^help ^objectspec' 以深入瞭解標籤

規格。)

變更集規格 標籤變更集的名稱或完整規格。

(使用 'cm ^help ^objectspec' 以深入瞭解變更集

規格。)

工作區\_路徑 標籤工作區的路徑。(工作區所指向的

變更集將加註標籤。)

選項：

--^allxlinkedrepositories 建立在所有 Xlink 連結儲存庫中的新標籤。

-^c 將指定的註解套用至新標籤。

-^commentsfile 將指定檔案中的註解套用至

新標籤。

== CMD\_HELP\_LABEL\_CREATE ==

備註：

設定 PLASTICEDITOR 環境變數，以指定要輸入

註解的編輯器。

範例：

cm ^label ^create ^lb:BL001 ^cs:1203 -^commentsfile=commentlb001.txt

(建立變更集 1203 所連接的標籤 'BL001'，並套用在

'commentlb001.txt' 檔案中的註解。)

cm ^label BL002 ^cs:1203 -^c="初次發行"

(建立標籤 'BL002'，含註解，並連接至變更集 1203。)

== CMD\_DESCRIPTION\_LABEL\_DELETE ==

刪除一或多個標籤。

== CMD\_USAGE\_LABEL\_DELETE ==

用法：

cm ^label ^delete <標籤規格>[ ...]

標籤規格 要刪除的標籤。使用空格分隔標籤。

(使用 'cm ^help ^objectspec' 以深入瞭解標籤

規格。)

== CMD\_HELP\_LABEL\_DELETE ==

備註：

此命令刪除一或多個標籤。

範例：

cm ^label ^delete ^lb:BL001

(刪除標籤 'BL001'。)

cm ^label ^delete ^lb:BL001 ^lb:BL002@reptest@server2:8084

(刪除標籤 'BL001' 和 'BL002'。)

== CMD\_DESCRIPTION\_LABEL\_RENAME ==

重新命名標籤。

== CMD\_USAGE\_LABEL\_RENAME ==

用法：

cm ^label ^rename <標籤規格> <新\_名稱>

標籤規格 要重新命名的標籤。

(使用 'cm ^help ^objectspec' 以深入瞭解標籤規格。)

新\_名稱 新增標籤的名稱。

== CMD\_HELP\_LABEL\_RENAME ==

備註：

此命令會重新命名標籤。

範例：

cm ^label ^rename ^lb:BL001 BL002

(將標籤 'BL001' 重新命名成 'BL002'。)

== CMD\_DESCRIPTION\_OBLITERATE ==

已棄用

== CMD\_USAGE\_OBLITERATE ==

已棄用。

== CMD\_HELP\_OBLITERATE ==

已棄用。

== CMD\_DESCRIPTION\_OBJECTSPEC ==

描述如何寫入物件規格。

== CMD\_USAGE\_OBJECTSPEC ==

用法：

cm ^objectspec

可取得所有關於如何建立物件規格的資訊。

== CMD\_HELP\_OBJECTSPEC ==

多種 Plastic SCM 命令都會將「物件規格」當做輸入，以便參考至

指定的物件 (通常是分支、變更集、儲存庫等)。

這份文件說明不同的可用「規格」，及其

建置方式。

每種規格類型都會以唯一標籤做為開頭，例如 "^rep:" 或 "^cs:"。這個標籤

必須專為接受一般物件規格的命令建立，例如，

"cm ^setowner 物件\_規格"，但只接受單一類型規格的命令，例如

"cm ^getfile 修訂\_規格"，經常會略過此標籤。

-- 儲存庫伺服器規格 (repserverspec) --

^repserver:name:port

範例：

cm ^repo ^list ^repserver:skull:8084

cm ^repo ^list skull:8084

快速筆記：

我們將此類規格稱為「儲存庫伺服器規格」，而不光只是

「伺服器規格」，原因如下。很久以前，我們有不同的工作區和儲存庫

伺服器，而沿用了當時的命名方式。

-- 儲存庫規格 (repspec) --

^rep:rep\_name@[儲存庫伺服器規格]

範例：

cm ^showowner ^rep:codice@localhost:6060

(在這個範例中，必須使用 "^rep:"，因為 ^showowner 不只准許儲存庫

也准許其他類型的物件。所以會要求使用者指明

物件類型。)

-- 分支規格 (brspec) --

^br:[/]br\_name[@repspec]

範例：

cm ^switch ^br:/main@^rep:plastic@^repserver:skull:9095

(這時並不需要 "^br:"、"^rep" 和 "^repserver"，因此

命令准許更短的形式：

"cm ^switch main@plastic@skull:9095"。)

cm ^find ^revisions "^where ^branch='^br:/main/task001'"

備註：

分支並不強制使用初始符號 '/'。我們習慣以 /main、

/main/task001 (依此類推) 方式，指定所有分支。如今，我們偏用

較短的格式 main、main/task001，使命令更為簡潔。

-- 變更集規格 --

^cs:cs\_number|cs\_GUID[@repspec]

變更集的號碼或 GUID 可進行指定。

範例：

cm ^ls /code --^tree=ae1390ed-7ce9-4ec3-a155-e5a61de0dc77@code@skull:7070

-- 標籤規格 (labelspec) --

^lb:lb\_name[@repspec]

範例：

cm ^switch ^lb:RELEASE2.0

cm ^switch ^lb:RELEASE1.4@myrep@MYSERVER:8084

-- 修訂規格 --

修訂規格分為多種類型：

^rev:item\_path[#(分支規格|變更集規格|標籤規格]

^rev:^serverpath:item\_path#(分支規格|變更集\_規格|標籤\_規格)

^rev:^revid:rev\_id[@rep\_spec]

^rev:^itemid:item\_id#(分支\_規格|變更集\_規格|標籤\_規格)

範例：

cm ^diff ^rev:readme.txt#^cs:19 ^rev:readme.txt#^cs:20

cm ^diff ^serverpath:/doc/readme.txt#^cs:19@myrepo \

^serverpath:/doc/readme.txt#^br:/main@myrepo@localhost:8084

cm ^cat ^revid:1230@^rep:myrep@^repserver:myserver:8084

-- 項目規格 --

^item:path

很少使用。

範例：

cm ^find ^revision "^where ^item='^item:.'"

-- 屬性規格 --

^att:att\_name[@repspec]

範例：

cm ^attribute ^set ^att:merged@code@doe:8084 ^cs:25@code@doe:8084 完成

-- 擱置規格 --

^sh:sh\_number[@repspec]

範例：

cm ^diff ^sh:2 ^sh:4

-- 工作區規格 --

^wk:name@clientmachine

很少使用，僅套用到與工作區相關的命令。可用於

依據名稱和機器 (而不依據路徑) 來指定工作區。

範例：

cm ^showselector ^wk:codebase@modok

快速筆記：

這些是過去使用 Plastic SCM 2.x 時所建立的規格，當時使用

「工作區伺服器」作為集中儲存工作區中繼資料

的方式。現在這些方法已因效能因素而棄用。

== CMD\_DESCRIPTION\_PARTIAL ==

在局部工作區中執行命令。

== CMD\_USAGE\_PARTIAL ==

用法：

cm ^partial <命令> [選項]

命令：

^configure

^add

^undo

^co | ^checkout

^unco | ^undocheckout

^ci | ^checkin

^mv | ^move

^rm | ^remove

^stb | ^switch

^upd | ^update

若要取得每個命令的詳細資訊，請執行：

cm ^partial <命令> --^usage

cm ^partial <命令> --^help

== CMD\_HELP\_PARTIAL ==

範例：

cm ^partial ^configure +/background-blue.png

cm ^partial ^update landscape-1024.png

cm ^partial ^checkin eyes-green.png eyes-black.png

== CMD\_DESCRIPTION\_PARTIAL\_ADD ==

將項目新增至版本控制。

== CMD\_USAGE\_PARTIAL\_ADD ==

用法：

cm ^partial ^add [-^R | -^r | --^recursive] [--^silent] [--^parents]

[--^ignorefailed] [--^skipcontentcheck] <項目\_路徑>[ ...]

項目\_路徑 要新增的項目。使用雙引號 (" ") 指定含空格的

路徑。使用空格分隔路徑。

使用 \* 以新增目前目錄的所有內容。

選項：

-^R 遞迴新增項目。

--^silent 不顯示任何輸出。

--^parents 包括作業中所指定項目的

上層目錄。

--^ignorefailed 如果無法新增項目，新增作業將會在沒有項目的情況下，

繼續進行新增作業。請注意：如果無法新增目錄

，則不會新增其內容。

--^skipcontentcheck 當擴充功能不足以將檔案設為文字

或二進位時，將預設為二進位，而不是

檢查內容來偵測類型。

== CMD\_HELP\_PARTIAL\_ADD ==

備註：

新增項目的要求：

- 要新增之項目的上層目錄必須是先前已新增的目錄。

範例：

cm ^partial ^add pic1.png pic2.png

(新增項目 'pic1.png' 和 'pic2.png'。)

cm ^partial ^add c:\workspace\picture.png

(在路徑 'c:\workspace' 中新增項目 'picture.png'。)

cm ^partial ^add -^R c:\workspace\src

(遞迴新增 'src'。)

cm ^partial ^add --^parents samples\design01.png

(新增 'design01.png' 檔案和 'samples' 上層資料夾。)

cm ^partial ^add -^R \*

(遞迴新增目前目錄的所有內容。)

== CMD\_DESCRIPTION\_PARTIAL\_CHECKIN ==

在此儲存庫中儲存變更。

== CMD\_USAGE\_PARTIAL\_CHECKIN ==

用法：

cm ^partial ^checkin | ^ci [<項目\_路徑>[ ...]]

[-^c=<字串\_註解> | -^commentsfile=<註解\_檔案>]

[--^all | -^a] [--^applychanged] [--^keeplock]

[--^symlink] [--^ignorefailed]

選項：

項目\_路徑 要簽入的項目。使用雙引號 (" ") 指定

含空格的路徑。使用空格分隔

路徑。

使用 . 將簽入套用至目前目錄。

-^c 指定簽入作業中所建立變更集

的註解。

-^commentsfile 將指定檔案的註解套用至

簽入作業中建立的變更集。

--^all | -^a 在指定路徑上，從本地進行變更、移動和刪除

的項目也包括在內。

--^applychanged 將簽入作業套用至工作區中偵測到

的變更項目，以及簽出

項目。

--^keeplock 在簽入作業後繼續保留鎖定已鎖定的

項目。

--^symlink 將簽入作業套用至符號連結，而不是

套用至目標。

--^ignorefailed 任何無法套用的變更 (因無法擷取鎖定

- 又名為獨佔簽出，或因本地變更

與伺服器變更存在衝突) 已遭捨棄，

因此簽入作業會繼續執行，

但不包含這些變更。

== CMD\_HELP\_PARTIAL\_CHECKIN ==

備註：

- 如果未指定 <項目\_路徑>，簽入將涉及工作區中

所有暫止的變更。

- 此簽入作業一律從指定路徑遞迴套用。

- 若要簽入項目：

- 項目必須受原始程式碼控制。

- 必須簽出項目。

- 如果項目已變更，但未簽出，則不一定要有 '--^applychanged' 標誌

，除非 <項目\_路徑> 是目錄，或包含

萬用字元 ('\*')。

修訂內容應該與先前修訂不同，才能

簽入。

設定 PLASTICEDITOR 環境變數，以指定要輸入

註解的編輯器。

從 stdin 讀取輸入：

命令 '^partial ^checkin' 可從 stdin 讀取路徑。若要執行此操作，可

傳遞單一破折號「-」來完成。

範例：cm ^partial ^checkin -

將讀取路徑，直到輸入空白行為止。

這可讓您使用管線以指定要簽入的檔案。

範例：

dir /S /B \*.c | cm ^partial ^checkin --^all -

(在 Windows 中，簽入工作區中所有 .c 檔案。)

範例：

cm ^partial ^checkin figure.png landscape.png

(套用簽入到 'figure.png' 和 'landscape.png' 已簽出檔案。)

cm ^partial ^checkin .-^commentsfile=mycomment.txt

(套用簽入到目前的目錄，並從 'mycomment.txt' 檔案設定

註解。)

cm ^partial ^ci background.png -^c="我的註解" --^keeplock

(套用簽入到 'background.png'，包括註解，並保留

鎖定狀態。)

cm ^partial ^checkin --^applychanged

(套用簽入到工作區中的所有暫止變更。)

cm ^partial ^checkin 連結 --^symlink

(套用簽入到連結檔案，而非目標；適用於

UNIX 環境。)

cm ^partial ^checkin .--^ignorefailed

(套用簽入到目前的目錄，因此忽略無法套用的

變更。)

== CMD\_DESCRIPTION\_PARTIAL\_CHECKOUT ==

將檔案標示為準備好修改。

== CMD\_USAGE\_PARTIAL\_CHECKOUT ==

用法：

cm ^partial ^checkout | ^co [<項目\_路徑>[ ...]] [--^resultformat=<字串\_格式>]

[--^silent] [--^ignorefailed]

選項：

項目\_路徑 要簽出的項目。使用雙引號 (" ") 指定

含空格的路徑。使用空格分隔

路徑。

使用 . 將簽出套用至目前目錄。

--^resultformat 以指定的格式擷取輸出

結果訊息。

--^silent 不顯示任何輸出。

--^ignorefailed 如果無法鎖定項目 (無法執行獨佔簽出)，

則會在沒有項目的情況下，

繼續進行簽出作業。

== CMD\_HELP\_PARTIAL\_CHECKOUT ==

備註：

若要簽出項目：

- 項目必須受原始程式碼控制。

- 必須簽入項目。

如果在伺服器上已設定鎖定 (lock.conf 存在)，那麼每次在路徑上

發生簽出時，Plastic 會檢查其是否符合任何規則，如果符合，

該路徑會處於獨佔簽出 (已鎖定)，如此一來，就沒有人

可以同時簽出。

您可以使用 'cm ^lock ^list' 以取得伺服器中的所有鎖定。

查看《管理指南》以瞭解鎖定的運作方式：

https://www.plasticscm.com/download/help/locking

範例：

cm ^partial ^checkout pic1.png pic2.png

(簽出檔案 'pic1.png' 和 'pic2.png'。)

cm ^partial ^co \*.png

(簽出所有 png 檔案。)

cm ^partial ^checkout .

(簽出目前目錄。)

cm ^partial ^checkout -^R c:\workspace\src

(遞迴簽出 'src' 資料夾。)

== CMD\_DESCRIPTION\_PARTIAL\_CONFIGURE ==

允許透過將項目載入工作區或從其中取消載入，進行工作區設定。

== CMD\_USAGE\_PARTIAL\_CONFIGURE ==

用法：

cm ^partial ^configure <+|-路徑>[ ...][--^silent] [--^ignorefailed]

[--^ignorecase] [--^restorefulldirs]

路徑 要載入或取消載入的路徑。使用雙引號 (" ")

指定含空格的路徑。使用空格分隔

路徑。

路徑開頭必須為 "/"。

選項：

--^silent 不顯示任何輸出。

--^ignorefailed 略過過程中的所有錯誤。不正確的路徑

不會造成命令停止執行。

--^ignorecase 忽略路徑的大小寫區分。搭配這個標誌時，'^configure'

將適用於 "/Data/Textures"，即使使用者是寫入

"/data/teXtures"。

--^restorefulldirs 重設無效的目錄設定 (發生在

非局部作業執行於局部工作區中時)。

在這個清單中的目錄將會完全設定

(完整檢查)，這表示進行更新時，這些目錄都將

自動下載全新內容。

這個作業不會下載任何檔案，而是僅

還原在局部工作區上的

目錄設定。

== CMD\_HELP\_PARTIAL\_CONFIGURE ==

備註：

此命令應該會執行遞迴作業。

範例：

cm ^partial ^configure +/landscape\_grey.png

(載入項目 'landscape\_grey.png'。)

cm ^partial ^configure -/landscape\_black.png

(取消載入項目 'landscape\_black.png'。)

cm ^partial ^configure +/soft -/soft/soft-black.png

(載入所有 'soft' 目錄下層項目，但 'soft-black.png' 除外。)

cm ^partial ^configure -/

(取消載入整個工作區。)

cm ^partial ^configure -/ +/

(載入整個工作區。)

cm ^partial ^configure -/figure-64.png --^ignorefailed

(取消載入 'figure-64.png' 項目，即使其已經取消載入。)

cm ^partial ^configure +/ --^restorefulldirs

(將所有目錄設成自動下載新內容。)

cm ^partial ^configure +/src/lib --^restorefulldirs

(僅將 '/src/lib' 及其子目錄設成自動下載

新內容。)

== CMD\_DESCRIPTION\_PARTIAL\_MOVE ==

移動或重新命名檔案或目錄。

== CMD\_USAGE\_PARTIAL\_MOVE ==

用法：

cm ^partial ^move | ^mv <來源\_路徑> <目的地\_路徑> [--^format=<字串\_格式>]

來源\_路徑 來源項目路徑。

目的地\_路徑 目的地項目路徑。

選項：

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

== CMD\_HELP\_PARTIAL\_MOVE ==

備註：

這個命令會移動或重新命名儲存庫中的項目。

變更也是在本機檔案系統中完成。

如果來源路徑是檔案，則目的地路徑可能為檔案或

目錄。在第一個情況中，將會重新命名檔案；否則，將

移動項目。

如果來源路徑是個目錄，目的地路徑必須是目錄。

要移動或重新命名的項目必須存在。

格式：

{0} 來源路徑。

{1} 目的地路徑。

範例：

cm ^partial ^move file.png file-blue.png

(重新命名項目。)

cm ^partial ^mv .\file-blue.png .\blueFiles

(將 'file-blue.png' 移到 'blueFiles'。)

cm ^partial ^move .\design .\marketing

(重新命名目錄。)

== CMD\_DESCRIPTION\_PARTIAL\_RM ==

從版本控制刪除檔案或目錄。

== CMD\_USAGE\_PARTIAL\_RM ==

用法：

cm ^partial ^remove | ^rm <項目\_路徑>[ ...][--^nodisk]

項目\_路徑 要移除的項目路徑。使用雙引號 (" ")

指定含空格的路徑。使用空格分隔

路徑。

選項：

--^nodisk 從版本控制移除，但將項目保留在磁碟上。

== CMD\_HELP\_PARTIAL\_RM ==

備註：

會從磁碟刪除項目。會從原始程式碼控制的上層目錄中

移除已移除的項目。

要求：

- 項目必須受原始程式碼控制。

範例：

cm ^partial ^remove src

(移除 'src'。如果 'src' 是目錄，結果將與下面命令相同：

cm ^partial ^remove -^R src。)

cm ^partial ^remove c:\workspace\pic01.png --^nodisk

(將 'pic01.png' 從版本控制中移除，但保留於磁碟中。)

== CMD\_DESCRIPTION\_PARTIAL\_SWITCH ==

將分支設定為工作分支。

== CMD\_USAGE\_PARTIAL\_SWITCH ==

用法：

cm ^switch <分支\_規格> [--^report | --^silent] [--^workspace=<路徑>]

(設定工作分支，並更新工作區。)

cm ^switch <分支\_規格> --^configure <+|-路徑>[ ...][--^silent]

[--^ignorefailed] [--^ignorecase] [--^workspace=<路徑>]

(設定工作分支，並使工作區設定的執行如同 'cm

^partial ^configure' 命令。)

分支\_規格 分支規格。(使用 'cm ^help ^objectspec' 以深入瞭解

分支規格。)

路徑 要載入或取消載入的路徑。使用雙引號 (" ")

指定含空格的路徑。使用空格

分隔路徑。路徑開頭必須為 "/"。

選項：

--^silent 不顯示任何輸出。

--^report 在命令完成時列印已套用變更

的清單。使用 '--^silent' 時將會覆寫這個設定。

這個選項僅適用於未指定 '--^configure' 選項

的情況。

--^configure 在工作區分支更新後

設定 (載入 / 取消載入項目) 工作區。查看 'cm ^partial

^configure --^help' 以瞭解如何指定這些要設定的

路徑。

--^ignorefailed 略過設定過程中所出現的所有錯誤。

不正確的路徑不會造成命令停止執行。

--^ignorecase 忽略路徑的大小寫區分。搭配這個標誌時，選項

'--^configure' 將適用於 "/Data/Textures"，即使使用者

寫入的是 "/data/teXtures"。

--^workspace=路徑 工作區所在位置的路徑。

== CMD\_HELP\_PARTIAL\_SWITCH ==

備註：

此命令可讓使用者更新工作分支。在更新

分支之後，此命令會將工作區更新成新分支，如同

'cm ^partial ^update' 命令所執行方式。不過，如果已指定 '--^configure' 選項，

此命令就可讓您使用新分支設定來

設定工作區，如同 'cm ^partial ^configure' 命令所執行方式。

範例：

cm ^switch ^br:/main/task

(將 /main/task 設為工作分支，並更新工作區。)

cm ^switch ^br:/main/task --^configure +/art/images

(將 /main/task 設為工作分支，並設定工作區可

載入 /art/images 資料夾。)

== CMD\_DESCRIPTION\_PARTIAL\_UNCO ==

復原項目的簽出。

== CMD\_USAGE\_PARTIAL\_UNCO ==

用法：

cm ^partial ^undocheckout | ^unco <項目\_路徑>[ ...][--^silent]

項目\_路徑 要套用作業的項目。使用雙引號 (" ")

指定含空格的路徑。使用空格

分隔路徑。

使用 . 將作業套用至目前目錄。

選項：

--^silent 不顯示任何輸出。

== CMD\_HELP\_PARTIAL\_UNCO ==

備註：

如果項目已簽出，而且您不想將其簽入，您可以使用此命令復原

簽出。檔案和資料夾皆可取消

簽出。項目將更新為簽出前的狀態。

要求：

- 項目必須受原始程式碼控制。

- 必須簽出項目。

範例：

cm ^partial ^undocheckout .

(復原目前目錄中的簽出。)

cm ^partial ^undocheckout pic1.png pic2.png

cm ^unco c:\workspace\design01.png

(復原所選檔案的簽出。)

== CMD\_DESCRIPTION\_PARTIAL\_UNDO ==

復原工作區中的變更。

== CMD\_USAGE\_PARTIAL\_UNDO ==

用法：

cm ^partial ^undo [<路徑>[ ...]] [--^symlink] [-^r | --^recursive]

[<篩選>[ ...]]

[--^silent | --^machinereadable [--^startlineseparator=<分隔符號>]

[--^endlineseparator=<分隔符號>] [--^fieldseparator=<分隔符號>]]

路徑 要套用作業之檔案或目錄的

路徑。使用雙引號 (" ") 指定

含空格的路徑。使用空格

分隔路徑。

如果未指定路徑，復原作業預設會

使用目前目錄中的

所有檔案。

篩選 將指定的一或多個篩選套用至指定的

路徑。使用空格分隔篩選。如需詳細資訊，請參閱

＜篩選＞一節。

選項：

--^symlink 將復原作業套用至符號連結，而不是

套用至目標。

-^r 以遞迴方式執行復原。

--^silent 不顯示任何輸出。

--^machinereadable 以易於剖析的格式輸出結果。

--^startlineseparator 搭配 '--^machinereadable' 標誌使用，指定

各行應如何起始。

--^endlineseparator 搭配 '--^machinereadable' 標誌使用，指定

各行應如何結束。

--^fieldseparator 搭配 '--^machinereadable' 標誌使用，指定

應如何分隔欄位。

篩選：

如果未指定標誌，會依預設復原所有變更，但

可使用下列一或多個標誌篩選路徑。

如果檔案或目錄符合一或多個指定的變更類型，

將會復原所述之檔案或目錄上的所有變更。

例如，如果您指定 '--^checkedout' 和 '--^moved'，若檔案

已簽出且移動，則會復原這兩項變更。

--^checkedout 選取已簽出的檔案和目錄。

--^unchanged 選取未變更內容的檔案。

--^changed 選取已在本地變更或簽出的檔案和

目錄。

--^deleted 選取已刪除的檔案和目錄。

--^moved 選取已移動的檔案和目錄。

--^added 選取已新增的檔案和目錄。

== CMD\_HELP\_PARTIAL\_UNDO ==

備註：

^undo 命令具有危險性 - 它會以無法回復的方式復原工作。

^undo 作業完成後，則無法復原受其影響的檔案和

目錄的先前狀態。如果未在引數中

指定路徑，預設將復原目前目錄中的所有變更，

但不是以遞迴方式進行。

這些等同於從 /src 目錄執行：

/src

|- file.txt

|- code.cs

\- /test

|- test\_a.py

\- test\_b.py

cm ^partial ^undo

cm ^partial ^undo \*

cm ^partial ^undo file.txt code.cs /test

cm ^partial ^undo .

cm ^partial ^undo /src file.txt code.cs

若要以遞迴的方式進行作業，您必須指定 '-^r' 標誌。

復原目錄下的所有變更 (包括影響到

目錄本身的變更)：

cm ^partial ^undo dirpath -^r

如果目錄路徑是工作區路徑，將會復原工作區中的

所有變更。

範例：

cm ^partial ^undo .-^r

(以遞迴方式復原目前目錄中的所有變更。如果已從

工作區的根目錄執行，則會復原整個工作區中的所有變更。)

cm ^partial ^co file.txt

cm ^partial ^undo file.txt

(復原 file.txt 的簽出。)

^echo ^content >> file.txt

cm ^partial ^undo file.txt

(復原 file.txt 的本地變更。)

cm ^partial ^undo src

(復原 src 目錄及其中檔案的變更。)

cm ^partial ^undo src/\*

(復原 src 中所包含各個檔案與目錄的變更，而

完全不影響 src。)

cm ^partial ^undo \*.cs

(復原在目前目錄中，與 \*.cs 相符的每個檔案或目錄的

變更。)

cm ^partial ^undo \*.cs -^r

(復原在目前目錄與底下每個目錄中，

與 \*.cs 相符的每個檔案或目錄。)

cm ^partial ^co file1.txt file2.txt

^echo ^content >> file1.txt

cm ^partial ^undo --^unchanged

(復原未變更 file2.txt 的簽出，因此忽略本地變更的

file1.txt。)

^echo ^content >> file1.txt

^echo ^content >> file2.txt

cm ^partial ^co file1.txt

cm ^partial ^undo --^checkedout

(復原已簽出檔案 file1.txt 中的變更，因此忽略 file2.txt，因為這個檔案

並未簽出。)

cm ^partial ^add file.txt

cm ^partial ^undo file.txt

(復原新增 file.txt 的動作，因而使其再次成為私人檔案。)

^rm file1.txt

^echo ^content >> file2.txt

cm ^partial ^add file3.txt

cm ^partial ^undo --^deleted --^added \*

(復原 file1.txt 刪除動作和 file3.txt 新增動作，而忽略 file2.txt

變更。)

== CMD\_DESCRIPTION\_PARTIAL\_UPDATE ==

更新局部的工作區，以及下載最新的變更。

== CMD\_USAGE\_PARTIAL\_UPDATE ==

用法：

cm ^partial ^update [<項目\_路徑>[ ...]] [--^changeset=<數目>]

[--^silent | --^report] [--^dontmerge]

項目\_路徑 要更新的項目。使用雙引號 (" ") 指定

含空格的路徑。使用空格分隔

路徑。

使用 . 將更新套用至目前目錄。

如未指定任何路徑，則目前局部的工作區

會完整獲得更新。

選項：

--^changeset 更新局部工作區成特定變更集。

--^silent 不顯示任何輸出。

--^report 在命令完成時列印已套用變更

的清單。使用 '--^silent' 時將會覆寫這個設定。

--^dontmerge 不會合併檔案衝突，只是略過。

其他的變更隨即適當套用。這個選項

可用於自動化作業，免除使用者互動。

== CMD\_HELP\_PARTIAL\_UPDATE ==

備註：

命令 '^partial ^update' 會更新過時的檔案。

此命令應該會執行遞迴作業。

使用 '--^changeset' 選項時，如果所有的指定路徑都是在同一 Xlink

下的檔案，則可在 Xlink 連結儲存庫中的指定變更集中，

搜尋要下載的版本。

範例：

cm ^partial ^update

(更新目前局部工作區的所有內容。)

cm ^partial ^update .

(更新所有目前目錄下層項目。)

cm ^partial ^update backgroud-blue.png

(更新項目 'backgroud-blue.png'。)

cm ^partial ^update soft\_black.png soft-grey.png

(更新項目 'soft\_black.png' 和 'soft-grey.png'。)

cm ^partial ^update src --^report

(更新所有 'src' 目錄下層項目，並於最後列印出

套用的變更清單。)

cm ^partial ^update src --^changeset=4

(將所有 'src' 目錄下層項目更新成其在變更集 4 中

所載入的內容。)

cm ^partial ^update xlink/first.png --^changeset=4

(將 'xlink/first.png' 項目更新成其在 Xlink 連結儲存庫變更集 4 中

所載入的內容。

== CMD\_DESCRIPTION\_PATCH ==

從規格中產生修補檔案，或將產生的修補檔案套用到目前

工作區。

== CMD\_USAGE\_PATCH ==

用法：

cm ^patch <來源\_規格> [<來源\_規格>] [--^output=<輸出\_檔案>]

[--^tool=<差異比對\_的\_路徑>]

產生修補檔案，其中包含分支、變更集的差異，

或變更集之間的差異。這個檔案也會追蹤

文字檔和二進位檔案的差異。

cm ^patch --^apply <修補\_檔案> [--^tool=<修補\_的\_路徑>]

允許在目前工作區中套用所產生修補檔案

的內容。

來源\_規格 變更集或分支的完整規格。(使用

'cm ^help ^objectspec' 以深入瞭解規格。)

輸出\_檔案 要儲存修補內容的檔案。如果未指定任何檔案，

修補內容將會列印到標準輸出。

修補\_檔案 要套用於目前工作區的修補檔案。

選項：

--^output 設定修補命令的輸出檔案。

--^tool 設定要使用的應用程式 (差異比對或修補)。

== CMD\_HELP\_PATCH ==

限制：

如果已有輸出修補檔案，該命令將不會覆寫此檔案。

套用修補時，如果修改的檔案不在磁碟上，

命令將不會在其中套用變更。

重要：

這個命令需要使用差異比對 (Diff) 和修補 (Patch) 工具，公開提供於

http://gnuwin32.sourceforge.net/packages/patch.htm 和

http://gnuwin32.sourceforge.net/packages/diffutils.htm

安裝完成之後，建議在 PATH 環境變數中

新增其路徑。

範例：

cm ^patch ^cs:4@default@localhost:8084

(在主控台上以統一格式列印出 cset 4 的差異。)

cm ^patch ^br:/main --^output=file.patch

(產生修補檔案及分支 "main" 的差異。)

cm ^patch ^br:/main --^output=file.patch --^tool=C:\gnu\diff.exe

(功能如前所述，但使用 custom exe。)

cm ^patch ^cs:2@default ^cs:4@default

(在主控台上以統一格式列印出 csets 2 和 4 之間的差異。)

cm ^patch --^apply file.patch --^tool=C:\gnu\patch.exe

(使用 custom exe，將 'file.patch' 中的修補套用到本地工作區。)

== CMD\_DESCRIPTION\_QUERY ==

執行 SQL 查詢。需要 SQL 儲存位置。

== CMD\_USAGE\_QUERY ==

用法：

cm ^query <sql\_命令> [--^outputfile=<輸出\_檔案>]

[--^solveuser=<欄\_名>[,...]]

[--^solvepath=<欄\_名>[,...]]

[--^columnwidth=<值>] [--^nocolumnname]

[--^columnseparator=<分隔符號>] [--^repository=<名稱>]

sql\_命令 要執行的 sql 查詢。

選項：

--^outputfile 將結果寫入輸出檔案。

--^solveuser 將指定的欄位設為 [使用者名稱] 欄位。查詢

解譯器會將這些欄位的資料假定

為使用者，並將因此嘗試進行解析。

--^solvepath 將指定的欄位設為 [項目 ID] 欄位。查詢解譯器

將嘗試將項目 ID 解析為檔案系統

路徑。

--^columnwidth 指定每個欄的寬度以格式化輸出。

--^nocolumnname 不列印欄名。

--^columnseparator 使用字元做為欄分隔符號，而不使用定位字元 (\t)。

--^repository 要查詢的儲存庫。

== CMD\_HELP\_QUERY ==

備註：

此命令可讓使用者在伺服器資料庫中執行 SQL 查詢。

為寫入 SQL 查詢，請使用下列兩個預先定義的函數來管理

使用者和路徑：

- '^SolveUser(<使用者名稱>)'：將使用者名稱解析為 Plastic SCM 格式。

- '^SolvePath(<路徑>)'：將磁碟路徑解析為項目 ID。

此外，您可以使用選項以人類可讀的格式顯示查詢結果。

您可以使用 '--^solveuser=<欄\_名>' 和

'--^solvepath=<欄\_名>' 選項，指定查詢解譯器

必須轉換為可讀文字的欄。您可以指定多個欄名，

並使用逗號分隔。

範例：

cm ^query "^SELECT \* ^FROM ^revision" --^columnwidth=25 --^repository=reptest

(從 'reptest' 儲存庫的 '修訂' 表格擷取資料)

cm ^query "^SELECT b.^sname ^as 分支\_名稱, o.^dtimestamp ^as 日期 ^from ^branch b, \

^object o, ^seid s ^where b.^iobjid=o.^iobjid ^and o.^fidowner=s.^iseidid ^and \

s.^scode='^SolveUser(john)'" --^outputfile=query.txt

(輸出至以擁有者 'john' 做為分支的檔案。)

cm ^query "^select r.^iobjid, r.^fiditem ^as 路徑, s.^scode ^as 使用者名稱 ^FROM \

^revision r, ^object o, ^seid s ^WHERE r.^iobjid=o.^iobjid ^and \

o.^fidowner=s.^iseidid ^and o.^dtimestamp>2014/04/25" \

--^solveuser=使用者名稱 --^solvepath=路徑 --^repository=reptest@server2:9095

(從所選儲存庫擷取所選資料。)

cm ^query "^SELECT \* ^FROM ^revision ^WHERE ^fiditem=^SolvePath(c:\mywkpath\info)"

(擷取路徑 'info' 的所有修訂資料。)

== CMD\_DESCRIPTION\_ATTRIBUTE\_DELETE ==

刪除一或多個屬性。

== CMD\_USAGE\_ATTRIBUTE\_DELETE ==

用法：

cm ^attribute | ^att ^delete | ^rm <屬性\_規格>[ ...]

屬性\_規格 要刪除的屬性。使用空格分隔

屬性。

(使用 'cm ^help ^objectspec' 以深入瞭解屬性

規格。)

== CMD\_HELP\_ATTRIBUTE\_DELETE ==

備註：

此命令會移除一或多個屬性。

範例：

cm ^attribute ^delete ^att:狀態

(刪除 '狀態' 屬性。)

cm ^att ^rm 狀態 ^att:整合@reptest@server2:8084

(刪除 '狀態' 和 '整合' 屬性。)

== CMD\_DESCRIPTION\_ATTRIBUTE\_UNSET ==

取消設定物件的屬性。

== CMD\_USAGE\_ATTRIBUTE\_UNSET ==

用法：

cm ^attribute | ^att ^unset <屬性\_規格> <物件\_規格>

屬性\_規格 屬性規格。(使用 'cm ^help ^objectspec'

以深入瞭解屬性規格。)

物件\_規格 要移除屬性的物件

規格。可在下列項目上設定屬性：分支、變更集、

擱置集、標籤、項目和修訂。

(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

== CMD\_HELP\_ATTRIBUTE\_UNSET ==

備註：

此命令會取消設定先前在物件上所設定的屬性。它

不會刪除屬性物件本身。

範例：

cm ^attribute ^unset ^att:狀態 ^br:/main/SCM105

(從 'main/SCM105' 分支移除 '狀態' 屬性實現。)

cm ^att ^unset ^att:整合@reptest@localhost:8084 ^cs:25@reptest@localhost:8084

(從變更集 25 移除 '整合' 屬性實現，全部都在

'reptest' 儲存庫中。)

== CMD\_DESCRIPTION\_ATTRIBUTE\_RENAME ==

重新命名屬性。

== CMD\_USAGE\_ATTRIBUTE\_RENAME ==

用法：

cm ^attribute | ^att ^rename <屬性\_規格> <新\_名稱>

屬性\_規格 要重新命名的屬性。(使用 'cm ^help ^objectspec' 以深入

瞭解屬性規格。)

新\_名稱 屬性的新名稱。

== CMD\_HELP\_ATTRIBUTE\_RENAME ==

備註：

此命令會重新命名屬性。

範例：

cm ^attribute ^rename ^att:狀態 狀況

(將 '狀態' 屬性重新命名為 '狀況'。)

== CMD\_DESCRIPTION\_ATTRIBUTE\_EDIT ==

編輯屬性的註解。

== CMD\_USAGE\_ATTRIBUTE\_EDIT ==

用法：

cm ^attribute | ^att ^edit <屬性\_規格> <新\_註解>

屬性\_規格 要變更其註解的屬性。(使用 'cm ^help ^objectspec'

以深入瞭解屬性規格。)

新\_註解 屬性的新註解。您也可以為屬性指定

預設的值清單。

== CMD\_HELP\_ATTRIBUTE\_EDIT ==

備註：

此命令會變更屬性的註解。

若要為屬性指定預設的值清單，您只需要在屬性註解中

加入類似以下的一行：

'預設: 第一個值, "第二個值", 第 3 個值, "最後一個值"'。

範例：

cm ^attribute ^edit ^att:狀態 "CI 管線中的分支狀態。"

(編輯 '狀態' 屬性的註解。)

cm ^attribute ^edit ^att:狀態 "分支狀態。預設：未解決、已解決、已檢閱"

(編輯 '狀態' 屬性的註解。也會指定值的

清單。因此，當您將 '狀態' 屬性設定為物件時，您可以選取

下列其中一個值："未解決"、"已解決" 或 "已檢閱"。)

== CMD\_DESCRIPTION\_REPLICATE ==

警告：此命令已棄用。

請使用 'cm ^pull' (等同於 '^replicate') 和 'cm ^push' (等同於

'^replicate --^push')。

== CMD\_USAGE\_REPLICATE ==

== CMD\_HELP\_REPLICATE ==

== CMD\_DESCRIPTION\_PULL ==

從另一個儲存庫提取分支。

== CMD\_USAGE\_PULL ==

用法：

cm ^pull <來源\_分支\_規格> <目的地\_儲存庫\_規格>

[--^preview] [--^nodata] [轉譯選項]

[--^user=<使用者\_名稱> [--^password=<密碼>] | 授權選項]

(直接 server-to-server 複寫。從儲存庫提取分支。)

cm ^pull <目的地\_儲存庫\_規格> --^package=<套件\_檔案> [授權選項]

(套件型複寫。在目的地儲存庫中匯入套件。)

cm ^pull ^hydrate <目的地\_分支\_規格> [<來源\_儲存庫\_規格>]

[--^user=<使用者\_名稱> [--^password=<密碼>] | 授權選項]

(引入先前

使用 '--^nodata' 複寫之分支的所有變更集的遺失資料。如果沒有指定要取得資料的儲存庫，

Plastic 會嘗試使用「複寫來源」(已複寫

分支的來源))。

cm ^pull ^hydrate <目的地\_變更集\_規格> [<來源\_儲存庫\_規格>]

[--^user=<使用者\_名稱> [--^password=<密碼>] | 授權選項]

(引入先前使用

'--^nodata' 複寫之變更集的遺失資料。如果沒有指定要取得資料的儲存庫，Plastic 會嘗試

使用「複寫來源」)。

來源\_分支\_規格 要從遠端儲存庫提取的分支。

(使用 'cm ^help ^objectspec' 以深入瞭解分支規格。)

目的地\_分支\_規格 要合成的分支。

(使用 'cm ^help ^objectspec' 以深入瞭解分支規格。)

目的地\_變更集\_規格 要合成的變更集。

(使用 'cm ^help ^objectspec' 以深入瞭解變更集

規格。)

目的地\_儲存庫\_規格 目的地儲存庫。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫

規格。)

--^package 指定要針對套件型複寫匯入的先前建立

的套件檔案。

在沒有直接網路連線的情況下，可用在

伺服器之間移動資料。

請參照 'cm ^push' 以建立套件檔案。

選項：

--^preview 提供有關將提取哪些變更的資訊，但實際上

不會執行任何變更。此選項很適合用於

在複寫變更前檢查將傳輸

的資料。

--^nodata 複寫分支變更，而不複寫

資料。此選項不可與套件

複寫搭配使用。

轉譯選項 如需詳細資訊，請參閱＜轉譯選項＞一節。

--^user, --^password 如果來源和目的地的驗證模式

不同且沒有可對目的地進行驗證的設定檔時，

所要使用的認證。

驗證選項 如需詳細資訊，請參閱

＜驗證選項＞一節。

轉譯選項：

--^trmode=(^copy|^name|^table --^trtable=<轉譯\_表格\_檔案>)

來源和目的地儲存庫可能會使用不同的驗證

模式。'--^trmode' 選項可指定如何將使用者名稱從

來源轉譯至目的地。'--^trmode' 必須是下列其中

一個值：

^copy (預設)。表示只會複製使用者識別碼。

^name 表示將按名稱比對使用者識別碼。

^table 使用 '--^trtable' 選項中指定的轉譯表格

(請參閱下方)。

--^trtable=<轉譯\_表格\_檔案>

如果轉譯模式為 '表格'，轉譯表格則為包含

<舊名稱;新名稱> (每行一個) 格式的文字行檔案。當

分支寫入至目的地儲存庫時，來源儲存庫中 '舊名稱'

所識別之使用者所建立的物件將設定至目的地上

具有 '新名稱' 的使用者。

驗證選項：

您可使用下列兩種模式之一指定驗證資料：

1) 使用驗證參數：--^authmode=<模式> --^authdata=<資料>

--^authmode=(^NameWorkingMode|^LDAPWorkingMode|^ADWorkingMode|^UPWorkingMode)

範例：

(^LDAPWorkingMode) --^authdata=::0:dave:fPBea2rPsQaagEW3pKNveA

(^UPWorkingMode) --^authdata=dave:fPBea2rPsQaagEW3pKNveA==

'--^authdata' 這一行是 client.conf 檔案和 profiles.conf 檔案中 <^SecurityConfig> 項目

的內容。您可從 Plastic GUI 的 [喜好設定] 下的複寫

設定檔索引標籤，輕鬆產生 profiles.conf

檔案。

如果您使用的是 ^UPWorkingMode，可以直接指定：

--^authmode=^UPWorkingMode --^user=<使用者> --^password=<密碼>

2) 驗證檔案，其中針對各個連線的目標伺服器您可能

有不同的檔案，包括該伺服器的認證。

--^authfile=<驗證\_檔案>

此檔案包含 2 行：

行 1) 是模式 (如 '--^authmode' 所述)

行 2) 是驗證資料 (如 '--^authdata' 所述)

== CMD\_HELP\_PULL ==

備註：

'^pull' 命令能夠在來源儲存庫與目的地儲存庫之間

複寫分支 (及其變更集)。

儲存庫可位於不同的伺服器。

複寫作業分成兩種：'^push' 和 '^pull'。

'^pull' 作業是指複寫作業會要求將

來自來源儲存庫的資料儲存至目的地儲存庫。

用戶端將連線至目的地儲存庫，並且將從該主機

建立與來源儲存庫的連線，以擷取

鎖定目標的資料。提取期間，它是將

連線至來源的目的地伺服器。

雖然在一般分散式案例中，開發人員會將資料從其本機伺服器

推送至主要伺服器，但開發人員可能也會想要從主要伺服器

提取最新儲存庫更新。

若對兩個複寫的儲存庫上的相同分支

同時進行了變更，複寫可以解決此類情況：

- 推送：如果您嘗試將資料推送至儲存庫，而當中的變更比您傳送的

變更要新，系統將要求您提取最新變更、解決合併作業，

最後再次嘗試推送。

- 提取：每當您從遠端分支提取變更集時，它們會直接連結至

其上層變更集。如果您提取的變更集

不是分支中最後一個變更集的下層項目，則會出現多標頭

案例。分支將有多個「標頭」，或在分支上有

最後一個變更集。您必須先合併兩個「標頭」，

才能再次推送。

提取可在兩種模式下運作：

1) 伺服器之間的直接通訊：目的地伺服器將從來源

伺服器擷取資料，自動為指定的分支

同步資料。

2) 使用推送和 '--^package' 選項匯入先前產生的套件。

模式 1) 會要求使用者執行命令以接受遠端伺服器的驗證，

無論是使用client.conf 檔案中的預設驗證、

指定 '--^authmode' 和 '--^authdata' 修飾詞或是

'--^authmode' 和 '--^user' 和 '--^password' (如果驗證模式為

^UPWorkingMode。)

模式 2) 會要求使用先前使用推送

命令產生的套件檔案。

請注意，提取複寫會以間接的方式運作。執行時，

命令會要求目的地儲存庫連線至來源並

取得所選分支。

不過，這可直接使用推送命令完成。

這會讓命令將所選分支從來源複寫至

目的地。

範例：

cm ^pull ^br:/main@project1@remoteserver:8084 projectx@myserver:8084

(將 'main' 分支從 'remoteserver' 提取至 'myserver'。在本例中，

會使用相同的驗證模式設定這兩個伺服器。)

cm ^pull ^br:/main@project1@remoteserver:8084 projectx@myserver:8084 \

--^authmode=^LDAPWorkingMode --^authdata=::0:dave:fPBea2rPsQaagEW3pKNveA

(與之前一樣，提取相同的分支，但現在遠端伺服器已設定為

透過 Active Directory 驗證使用者。例如，我正從 Linux 電腦

連線至設定為使用 Active Directory 整合模式的

Windows 伺服器。我會指定我的 Active Directory 使用者和加密的

密碼，並以 LDAP 形式將它傳遞至伺服器。)

cm ^pull ^br:/main@project1@remoteserver:8084 projectx@myserver:8084 \

--^authmode=^UPWorkingMode --^user=dave --^password=mysecret

(提取相同的分支，但現在使用者是在遠端伺服器上進行驗證，

善加利用

Plastic SCM 所包含的使用者/密碼資料庫。)

cm ^pull ^br:/main@project1@remoteserver:8084 projectx@myserver:8084 --^nodata

(將 'main' 分支從 'remoteserver' 複寫至 'myserver'，但不包含資料。)

cm ^pull ^hydrate ^br:/main@projectx@myserver:8084 projectx@remoteserver:8084

(透過從遠端伺服器取得資料，

合成 'main' 分支中的所有變更集。)

cm ^pull ^hydrate ^cs:122169@projectx@myserver:8084 projectx@remoteserver:8084

(透過從遠端伺服器取得資料，

，合成 'myserver' 中的變更集 122169。)

== CMD\_DESCRIPTION\_PUSH ==

將分支推送至另一個儲存庫。

== CMD\_USAGE\_PUSH ==

用法：

cm ^push <來源\_分支\_規格> <目的地\_儲存庫\_規格>

[--^preview] [轉譯選項]

[--^user=<使用者\_名稱> [--^password=<密碼>] | 授權選項]

(直接 server-to-server 複寫。從儲存庫推送分支。)

cm ^push <來源\_分支\_規格> --^package=<套件\_檔案> [授權選項]

(套件型複寫。在包含所選分支的來源伺服器中

建立複寫套件。)

來源\_分支\_規格 要推送至遠端儲存庫的分支。

(使用 'cm ^help ^objectspec' 以深入瞭解分支規格。)

目的地\_儲存庫\_規格 目的地儲存庫。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫

規格。)

--^package 指定要針對套件型複寫匯出

複寫套件的路徑。

在沒有直接網路連線的情況下，

可用在伺服器之間移動資料。

選項：

--^preview 提供有關將推送哪些變更的資訊，

但實際上不會執行任何變更。此選項

很適合用來在複寫變更前

檢查將傳輸的資料。

轉譯選項 如需詳細資訊，請參閱＜轉譯選項＞一節。

--^user, --^password 如果來源和目的地的驗證模式

不同且沒有可對目的地進行驗證的設定檔時，

所要使用的認證。

驗證選項 如需詳細資訊，請參閱

＜驗證選項＞一節。

轉譯選項：

--^trmode=(^copy|^name|^table --^trtable=<轉譯\_表格\_檔案>)

來源和目的地儲存庫可能會使用不同的驗證

模式。'--^trmode' 選項可指定如何將使用者名稱

從來源轉譯至目的地。'--^trmode' 必須是下列

其中一個值：

^copy (預設)。表示只會複製使用者識別碼。

^name 表示將按名稱比對使用者識別碼。

^table 使用 '--^trtable' 選項中指定的轉譯表格

(請參閱下方)。

--^trtable=<轉譯\_表格\_檔案>

如果轉譯模式為 '表格'，轉譯表格則為包含

<舊名稱;新名稱> (每行一個) 格式的文字行檔案。當

分支寫入至目的地儲存庫時，來源儲存庫中 '舊名稱'

所識別之使用者所建立的物件將設定至目的地上

具有 '新名稱' 的使用者。

驗證選項：

您可使用下列兩種模式之一指定驗證資料：

1) 使用驗證參數：--^authmode=<模式> --^authdata=<資料>

--^authmode=(^NameWorkingMode|^LDAPWorkingMode|^ADWorkingMode|^UPWorkingMode)

範例：

(^LDAPWorkingMode) --^authdata=::0:dave:fPBea2rPsQaagEW3pKNveA

(^UPWorkingMode) --^authdata=dave:fPBea2rPsQaagEW3pKNveA==

'--^authdata' 這一行是 client.conf 檔案和 profiles.conf 檔案中 <^SecurityConfig> 項目

的內容。您可從 Plastic GUI 的 [喜好設定] 下的複寫

設定檔索引標籤，輕鬆產生 profiles.conf

檔案。

如果您使用的是 ^UPWorkingMode，可以直接指定：

--^authmode=^UPWorkingMode --^user=<使用者> --^password=<密碼>

2) 驗證檔案，其中針對各個連線的目標伺服器您可能

有不同的檔案，包括該伺服器的認證。

--^authfile=<驗證\_檔案>

此檔案包含 2 行：

行 1) 是模式 (如 '--^authmode' 所述)

行 2) 是驗證資料 (如 '--^authdata' 所述)

== CMD\_HELP\_PUSH ==

備註：

'^push' 命令能夠在來源儲存庫與目的地儲存庫之間

複寫分支 (及其變更集)。

儲存庫可位於不同的伺服器。

複寫作業分成兩種：'^push' 和 '^pull'。

'^push' 作業是指複寫作業會將資料

從來源儲存庫傳送至目的地儲存庫。在此案例中，

用戶端將連線至來源儲存庫，取得要複寫的資料，

然後將該資料傳送至目的地儲存庫。前者 (來源)

必須連線至目的地，而後者 (目的地)

不會將本身連線至來源。

在一般的分散式案例中，開發人員會將資料從其本機伺服器

推送至主要伺服器。此外，開發人員可能也會想要從主要伺服器

提取最新的儲存庫更新。

若對兩個複寫的儲存庫上的相同分支同時

進行了變更，複寫可以解決此類情況。

- 推送：如果您嘗試將資料推送至儲存庫，而當中的變更比您傳送

的變更要新，系統將要求您提取最新變更、解決合併作業，

最後再次嘗試推送。

- 提取：每當您從遠端分支提取變更集時，它們會直接連結至

其上層變更集。如果您提取的變更集

不是分支中最後一個變更集的下層項目，則會出現多標頭

案例。分支將有多個「標頭」，或在分支上有

最後一個變更集。您必須先合併兩個「標頭」，

才能再次推送。

推送可在兩種模式下運作：

1) 伺服器之間的直接通訊：原始伺服器會將資料

傳送至目的地伺服器，自動為

指定的分支同步資料。

2) 匯出套件模式：用戶端只會連線至來源並

產生複寫套件，取得指定分支的

資料和中繼資料。不會使用 '--^package' 修飾詞。

這兩種模式都會要求使用者執行命令以接受

伺服器的驗證，無論是使用 client.conf

檔案中的預設驗證或是指定 '--^authmode' 和 '--^authdata' 修飾詞。

^push 複寫會以直接的方式運作。執行時，此命令

會將所選分支從來源複寫至目的地，而不

要求目的地儲存庫連線至來源和取得

所選分支 (如同提取所執行的動作)。

範例：

cm ^push ^br:/main@project1@myserver:8084 projectx@remoteserver:8084

(將 'main' 分支從 'myserver' 複寫至 'remoteserver'。在本例中，

會使用相同的驗證模式設定這兩個伺服器。)

cm ^push ^br:/main@project1@myserver:8084 projectx@remoteserver:8084 \

--^authmode=^LDAPWorkingMode --^authdata=::0:dave:fPBea2rPsQaagEW3pKNveA

(與之前一樣，複寫相同的分支，但現在遠端伺服器已設定為

透過 Active Directory 驗證使用者。例如，我正從 Linux 電腦

連線至設定為使用 Active Directory 整合模式的

Windows 伺服器。我會指定我的 Active Directory 使用者和加密的

密碼，並以 LDAP 形式將它傳遞至伺服器。)

cm ^push ^br:/main@project1@myserver:8084 projectx@remoteserver:8084 \

--^authmode=^UPWorkingMode --^user=dave --^password=mysecret

(複寫相同的分支，但現在使用者是在遠端伺服器上進行驗證，

善加利用

Plastic SCM 所包含的使用者/密碼資料庫。)

== CMD\_DESCRIPTION\_CLONE ==

複製遠端儲存庫。

== CMD\_USAGE\_CLONE ==

用法：

cm ^clone <來源\_儲存庫\_規格> [<目的地\_儲存庫\_規格> | <目的地\_儲存庫伺服器\_規格>]

[--^user=<使用者\_名稱> [--^password=<密碼>] | 授權選項]

[轉譯選項]

(直接 repository-to-repository 複製。)

cm ^clone <來源\_儲存庫\_規格> --^package=<套件\_檔案>

[--^user=<使用者\_名稱> [--^password=<密碼>] | 授權選項]

(複製至中繼套件，之後可透過

提取至目的地儲存庫的方式匯入。)

來源\_儲存庫\_規格 複製作業的來源儲存庫。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫

規格。)

目的地\_儲存庫\_規格 複製作業的目的地儲存庫。如果存在

儲存庫，則它必須是空的。如果儲存庫不存在，則會

建立儲存庫。

如果未指定儲存庫，此命令會使用使用者的

預設儲存庫伺服器。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫

規格。)

目的地\_儲存庫伺服器\_規格 複製作業的目的地儲存庫伺服器。

如果有一個儲存庫的名稱與

目的地儲存庫伺服器中的 <來源\_儲存庫\_規格> 相同，則它

必須是空的。如果沒有儲存庫，則會建立儲存庫。

如果未指定儲存庫，此命令會使用使用者的

預設儲存庫伺服器。

(使用 'cm ^help ^objectspec' 以深入瞭解儲存庫

伺服器規格。)

選項：

--^user, --^password 如果來源和目的地的驗證模式

不同且沒有可對目的地進行驗證的設定檔時，

所要使用的認證。

--^package 將指定的儲存庫匯出至套件檔案，

而不是匯出至儲存庫。

在沒有直接網路連線的情況下，可用在

伺服器之間移動資料。

產生的套件必須透過提取命令

匯入。

轉譯選項 如需詳細資訊，請參閱＜轉譯選項＞一節。

驗證選項 如需詳細資訊，請參閱

＜驗證選項＞一節。

轉譯選項：

--^trmode=(^copy|^name|^table --^trtable=<轉譯\_表格\_檔案>)

來源和目的地儲存庫可能會使用不同的驗證

模式。'--^trmode' 選項可指定如何將使用者名稱從

來源轉譯至目的地。'--^trmode' 必須是下列其中

一個值：

^copy (預設)。表示只會複製使用者識別碼。

^name 表示將按名稱比對使用者識別碼。

^table 使用 '--^trtable' 選項中指定的轉譯表格

(請參閱下方)。

--^trtable=<轉譯\_表格\_檔案>

如果轉譯模式為 '表格'，轉譯表格則為包含

<舊名稱;新名稱> (每行一個) 格式的文字行檔案。當

分支寫入至目的地儲存庫時，來源儲存庫中 '舊名稱'

所識別之使用者所建立的物件將設定至目的地上

具有 '新名稱' 的使用者。

驗證選項：

您可使用下列兩種模式之一指定驗證資料：

1) 使用驗證參數：--^authmode=<模式> --^authdata=<資料>

--^authmode=(^NameWorkingMode|^LDAPWorkingMode|^ADWorkingMode|^UPWorkingMode)

範例：

(^LDAPWorkingMode) --^authdata=::0:dave:fPBea2rPsQaagEW3pKNveA

(^UPWorkingMode) --^authdata=dave:fPBea2rPsQaagEW3pKNveA==

'--^authdata' 這一行是 client.conf 檔案和 profiles.conf 檔案中 <^SecurityConfig> 項目

的內容。您可從 Plastic GUI 的 [喜好設定] 下的複寫

設定檔索引標籤，輕鬆產生 profiles.conf

檔案。

如果您使用的是 ^UPWorkingMode，可以直接指定：

--^authmode=^UPWorkingMode --^user=<使用者> --^password=<密碼>

2) 驗證檔案，其中針對各個連線的目標伺服器您可能

有不同的檔案，包括該伺服器的認證。

--^authfile=<驗證\_檔案>

此檔案包含 2 行：

行 1) 是模式 (如 '--^authmode' 所述)

行 2) 是驗證資料 (如 '--^authdata' 所述)

== CMD\_HELP\_CLONE ==

備註：

複製命令能夠將分支 (及其變更集、標籤、屬性、檢閱

等項目) 從來源儲存庫複寫至目的地

儲存庫。儲存庫可位於不同的伺服器。

可事先建立目的地儲存庫，但如果它包含

先前的資料，複製作業將會失敗。

複製作業不會複製儲存庫子模組，也不會複製

Xlink 下的儲存庫。

範例：

cm ^clone awesomeProject@tardis@cloud

(將 'awesomeProject' 儲存庫從 'tardis@cloud' 組織複製至

具有相同名稱的本機儲存庫。)

cm ^clone repo@server.home:9095 repo-local

(將 'repo' 從 'server.home:9095' 複製至使用者的預設

儲存庫伺服器的 'repo-local'。)

cm ^clone project@192.168.111.130:8084 ^repserver:192.168.111.200:9095

(將 'project' 儲存庫從 '192.168.111.130:8084' 複製至

'project@192.168.111.200:9095'。)

cm ^clone project@ldapserver:8084 --authfile=credentials.txt \

--^trmode=table --^trtable=table.txt

(根據遠端儲存庫使用驗證檔案

從 'ldapserver:8084' 複製 'project' 儲存庫，並按照指定的轉譯表格

轉譯使用者。)

cm ^clone project@server.home:9095 --^package=project.plasticpkg

cm ^repository ^create project@mordor.home:8084

cm ^pull --^package=project.plasticpkg project@mordor.home:8084

(將 'project' 儲存庫從 'server.home:9095' 複製至

'project.plasticpkg' 套件，這之後會透過提取至

'mordor.home:8084' 的 'porject' 儲存庫的方式匯入。)

== CMD\_DESCRIPTION\_REVERT ==

將項目還原為先前修訂。

== CMD\_USAGE\_REVERT ==

用法：

cm ^revert <修訂規格>

修訂規格 包含將在工作區中載入之內容的修訂的

變更集規格。

(使用 'cm ^help ^objectspec' 以深入瞭解修訂

規格。)

== CMD\_HELP\_REVERT ==

備註：

必須簽入項目。

範例：

cm ^revert dir#^cs:0

cm ^revert C:\mywks\dir\file1.txt#23456

== CMD\_DESCRIPTION\_REVISION\_HISTORY ==

顯示檔案或目錄的歷程記錄。

== CMD\_USAGE\_REVISION\_HISTORY ==

用法：

cm ^history | ^hist <項目\_路徑>[ ...][--^long | --^format=<字串\_格式>]

[--^symlink] [--^xml[=<輸出\_檔案>]] [--^encoding=<名稱>]

項目\_路徑 項目路徑。使用空格分隔路徑。使用

雙引號 (" ") 指定含空格的路徑。

路徑也可以是伺服器路徑修訂。

(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

選項：

--^long 顯示其他資訊。

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

--^symlink 將歷程記錄作業套用至符號連結，而不是

套用至目標。

--^xml 將 XML 格式的輸出列印至標準輸出。

可以指定輸出檔案。

--^encoding 與 '--^xml' 選項搭配使用，指定要在

XML 輸出中使用的編碼，亦即：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

== CMD\_HELP\_REVISION\_HISTORY ==

備註：

此命令會顯示指定項目的修訂清單和標籤、分支，

以及每個修訂的註解資訊。

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{0} | {^date} 日期。

{1} | {^changesetid} 變更集號碼。

{2} | {^branch} 分支。

{4} | {^comment} 註解。

{5} | {^owner} 擁有者。

{6} | {^id} 修訂 ID。

{7} | {^repository} 儲存庫。

{8} | {^server} 伺服器。

{9} | {^repspec} 儲存庫規格。

{^tab} 插入定位字元空格。

{^newline} 插入新行。

範例：

cm ^history file1.txt "file 2.txt"

cm ^hist c:\workspace --^long

(顯示所有資訊。)

cm ^history 連結 --^symlink

(將歷程記錄作業套用至 '連結' 檔案，而非套用至目標；

此功能適用於 UNIX 環境。)

cm ^history ^serverpath:/src/foo/bar.c#^br:/main/task001@myserver

(從指定分支的伺服器路徑中擷取修訂版本歷程記錄。)

== CMD\_DESCRIPTION\_REVISION\_TREE ==

顯示項目的修訂樹。

== CMD\_USAGE\_REVISION\_TREE ==

用法：

cm ^tree <路徑> [--^symlink]

路徑 項目路徑。

選項：

--^symlink 將作業套用至連結檔案，而不是套用至目標。

== CMD\_HELP\_REVISION\_TREE ==

範例：

cm ^tree fichero1.txt

cm ^tree c:\workspace

cm ^tree 連結 --^symlink

(將作業套用至連結檔案，而不是套用至目標；此功能適用於

UNIX 環境。)

== CMD\_DESCRIPTION\_RM ==

可讓使用者刪除檔案和目錄。

== CMD\_USAGE\_RM ==

用法：

cm ^remove | ^rm <命令> [選項]

命令：

^controlled (選用)

^private

若要取得每個命令的詳細資訊，請執行：

cm ^remove <命令> --^usage

cm ^remove <命令> --^help

== CMD\_HELP\_RM ==

範例：

cm ^remove \path\controlled\_file.txt

cm ^remove ^private \path\private\_file.txt

== CMD\_DESCRIPTION\_RM\_CONTROLLED ==

從版本控制刪除檔案或目錄。

== CMD\_USAGE\_RM\_CONTROLLED ==

用法：

cm ^remove | ^rm <項目\_路徑>[ ...][--^format=<字串\_格式>]

[--^errorformat=<字串\_格式>] [--^nodisk]

項目\_路徑 要移除的項目路徑。使用雙引號 (" ") 指定

含空格的路徑。使用空格分隔

路徑。

選項：

--^format 以指定的格式擷取輸出進度

訊息。如需詳細資訊，請參閱範例。

--^errorformat 以指定的格式擷取錯誤訊息

(如果有的話)。如需詳細資訊，請參閱範例。

--^nodisk 從版本控制移除，但將項目保留在

磁碟上。

== CMD\_HELP\_RM\_CONTROLLED ==

備註：

會從磁碟刪除項目。會從原始程式碼控制的上層目錄中

移除已移除的項目。

要求：

- 項目必須受原始程式碼控制。

從 stdin 讀取輸入：

'^remove' 命令可從 stdin 讀取路徑。若要執行此操作，可傳遞單一

破折號「-」來完成。

範例：cm ^remove -

將讀取路徑，直到輸入空白行為止。

這可讓您使用管線指定要移除的檔案。

範例：

dir /S /B \*.c | cm ^remove -

(在 Windows 中，移除工作區中的所有 .c 檔案。)

範例：

cm ^remove src

(移除 'src'。如果 'src' 是目錄，這會與

'cm ^remove -^R src' 相同。)

cm ^remove c:\workspace\file.txt --^format="{0} - 已移除" \

--^errorformat="{0} - 移除時發生錯誤"

(從版本控制和磁碟中移除 'file.txt'；如果作業成功，則寫入

"c:\workspace\file.txt - ^REMOVED"，或如果作業失敗，則寫入

"c:\workspace\file.txt - ^ERROR ^REMOVING"。)

cm ^remove c:\workspace\file.txt --^nodisk

(從版本控制移除 'file.txt'，但將它保留在磁碟上。)

== CMD\_DESCRIPTION\_RM\_PRIVATE ==

刪除私人檔案或目錄。

警告：使用此命令刪除的檔案會永久刪除，而且無法

復原。建議您使用 '--^dry-run' 選項檢查

哪些檔案會受到此命令影響。

== CMD\_USAGE\_RM\_PRIVATE ==

用法：

cm ^remove | ^rm ^private <路徑>[ ...][-^R | -^r | --^recursive] [--^ignored]

[--^verbose] [--^dry-run]

路徑 要移除的檔案或目錄的路徑。使用雙

引號 (" ") 指定含空格的路徑。使用

空格分隔路徑。

選項：

--^r 以遞迴的方式從受控目錄內

刪除私人檔案。

--^ignored 同時刪除忽略和隱匿的檔案和目錄。

--^verbose 列印所有受影響的路徑。

--^dry-run 執行此命令，而不在磁碟上進行任何變更。

== CMD\_HELP\_RM\_PRIVATE ==

備註：

如果路徑是私人檔案或目錄，則會從磁碟中將其刪除。

如果路徑是受控檔案，此命令會失敗。

如果路徑是受控目錄，除非您

指定 '-^r' 選項 (如此一來，將會刪除指定目錄下的所有私人檔案和

目錄)，否則此命令會失敗。

範例：

cm ^remove ^private 私人\_目錄

(刪除 '私人\_目錄'。)

cm ^remove ^private c:\workspace\受控\_目錄

(失敗，因為 '受控\_目錄' 不是私人目錄。)

cm ^remove ^private -^r c:\workspace\受控\_目錄

(刪除 '受控\_目錄' 下的所有私人檔案和目錄。)

cm ^rm ^private --^dry-run --^verbose c:\workspace\受控\_目錄 -^r

(顯示受到刪除 '受控\_目錄' 下的私人檔案所影響的

所有路徑 (實際上未刪除任何項目)。)

cm ^rm ^private --^verbose c:\workspace\受控\_目錄 -^r

(顯示受到刪除 '受控\_目錄' 下的私人檔案所影響的

所有路徑 (執行刪除動作)。)

== CMD\_DESCRIPTION\_TRIGGER\_DELETE ==

刪除觸發器。

== CMD\_USAGE\_TRIGGER\_DELETE ==

用法：

cm ^trigger | ^tr ^delete | ^rm <子類型-類型> <位置\_號碼>

[--^server=<儲存庫伺服器規格>]

子類型-類型 觸發器執行與觸發器作業。

輸入 'cm ^showtriggertypes' 以查看觸發器

類型的清單。

位置\_號碼 建立觸發器時指派給它的位置。

選項：

--^server 在指定的伺服器上刪除觸發器。

如果未指定任何伺服器，請在用戶端上設定的

伺服器上執行命令。

== CMD\_HELP\_TRIGGER\_DELETE ==

範例：

cm ^trigger ^delete ^after-setselector 4

cm ^tr ^rm ^after-setselector 4

== CMD\_DESCRIPTION\_ATTRIBUTE\_SET ==

在指定的物件上設定屬性。

== CMD\_USAGE\_ATTRIBUTE\_SET ==

用法：

cm ^attribute | ^att ^set <屬性\_規格> <物件\_規格> <屬性\_值>

屬性\_規格 屬性規格。(使用 'cm ^help ^objectspec'

以深入瞭解屬性規格。)

物件\_規格 要設定屬性的物件規格。

可在下列項目上設定屬性：分支、變更集、

擱置集、標籤、項目和修訂。

(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

屬性\_值 要對物件設定的屬性值。

== CMD\_HELP\_ATTRIBUTE\_SET ==

備註：

可在物件上設定屬性，以儲存此物件的

其他資訊。

可在下列物件上設定屬性：分支、變更集、

擱置集、標籤、項目和修訂。

範例：

cm ^attribute ^set ^att:狀態 ^br:/main/SCM105 未解決

(使用 '未解決' 值將 '狀態' 屬性設定為 'SCM105' 分支。)

cm ^att ^set ^att:整合@reptest@server2:8084 ^lb:LB008@reptest@server2:8084 是

(在 'reptest' 儲存庫中以 '是' 值將 '整合' 屬性設定為

標籤 'LB008' 。)

== CMD\_DESCRIPTION\_SETOWNER ==

設定物件的擁有者。

== CMD\_USAGE\_SETOWNER ==

用法：

cm ^setowner | ^sto --^user=<使用者\_名稱> | --^group=<群組> <物件\_規格>

--^user 使用者名稱。物件的新擁有者。

--^group 群組名稱。物件的新擁有者。

物件\_規格 要設定新擁有者的物件規格。

可在下列物件上設定擁有者：

儲存庫伺服器、儲存庫、分支、變更集、

標籤、項目、修訂和屬性。

(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

== CMD\_HELP\_SETOWNER ==

備註：

物件擁有者可為使用者或群組。

可在下列物件上設定擁有者：儲存庫伺服器、

儲存庫、分支、變更集、標籤、項目、修訂和屬性。

範例：

cm ^setowner --^user=john ^repserver:localhost:8084

(將 'john' 設定為儲存庫伺服器擁有者。)

cm ^sto --^group=development ^rep:mainRep@PlasticServer:8084

(將 'development' 群組設定為 'mainRep' 儲存庫的擁有者。)

== CMD\_DESCRIPTION\_SETSELECTOR ==

將選擇器設定為工作區。

== CMD\_USAGE\_SETSELECTOR ==

此命令已棄用。它的存在是為了回溯相容性，

但在 Plastic SCM 4.0 中大多已棄用選擇器。選擇器仍

存在是為了指定工作分支或變更集，但舊規則

篩選路徑的功能已不再受到支援。

用法：

cm ^setselector | ^sts [--^file=<選擇器\_檔案>] [--^ignorechanges]

[--^forcedetailedprogress] [<工作區\_路徑> | <工作區\_規格>]

選項：

--^file 要從中載入選擇器的檔案。

--^ignorechanges 如果在更新工作區時偵測到

暫止的變更，則忽略顯示的暫止

變更警告訊息。

--^forcedetailedprogress 強制顯示詳細進度，即使在標準輸出

重新導向的情況下亦然。

工作區\_路徑 要設定選擇器的工作區路徑。

工作區\_規格 工作區規格。(使用 'cm ^help ^objectspec'

以深入瞭解工作區規格。)

== CMD\_HELP\_SETSELECTOR ==

備註：

此命令會設定工作區的選擇器。

工作區需要資訊才能從儲存庫載入修訂。

為取得此資訊，Plastic SCM 會使用選擇器。

透過選擇器，您可以從指定的分支、

標籤或變更集載入修訂。

如果未指定要載入選擇器的檔案，則會執行預設的作業

系統編輯器。

範例選擇器：

^repository "^default" // 工作中儲存庫

^path "/" // 規則將套用至根目錄

^branch "/^main" // 從 ^br:/^main 取得最新修訂

^checkout "/^main" // 將簽出置於分支 ^br:/^main 上

範例：

cm ^sts

(開啟要套用的目前選擇器檔案。)

cm ^sts ^wk:workspace\_projA@reptest

(開啟要套用的指定選擇器檔案。)

cm ^setselector --^file=c:\selectors\sel.xml

(在目前工作區中設定指定的選擇器檔案。)

cm ^setselector --^file=c:\selectors\sel.xml ^wk:MyWorkspace

(在所選工作區中設定指定的選擇器檔案。)

== CMD\_DESCRIPTION\_SHELVE ==

擱置已簽出項目的內容。

== CMD\_USAGE\_SHELVE ==

此命令已棄用。請改用 'cm ^shelveset'。

用法：

cm ^shelve [<項目\_路徑>+] [--^all] [--^dependencies]

[-^c=字串\_註解 | -^commentsfile=<註解\_檔案>]

[--^encoding=名稱] [--^comparisonmethod=比較\_方法]

(擱置內容。)

cm ^shelve --^apply=<擱置\_規格> [--^mount]

(套用已儲存的擱置集。)

--^apply 還原指定擱置集的擱置內容。

擱置規格：查看 'cm ^help ^objectspec'。

cm ^shelve --^delete=<擱置\_規格>

(移除已儲存的擱置集。)

--^delete 移除指定的擱置集。

擱置集規格：查看 'cm ^help ^objectspec'。

選項：

項目\_路徑 要擱置的項目 (以空格分隔)。可使用引號 (")

來指定含空格的路徑。

--^all 指定路徑上在本地變更、移動和刪除的項目

也會包括在內。

--^dependencies 將本地變更相依性包含在

要擱置的項目中。

-^c 將指定註解套用至已建立的擱置集。

-^commentsfile 將指定檔案中的註解套用至已建立的

擱置集。

--^encoding 指定輸出編碼，例如：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

--^comparisonmethod 設定比較方法。如需詳細資訊，請參閱備註。

--^mount 指定儲存庫的裝載點。

== CMD\_HELP\_SHELVE ==

備註：

如果未指定 <項目\_路徑> 也未指定任何選項，擱置將包含

工作區中所有暫止的變更。

擱置作業一律會從指定路徑以遞迴方式套用。

擱置項目的要求：

- 項目必須受原始程式碼控制。

- 項目必須簽出或變更 (必須使用 --^all 選項)。

比較方法:

^ignoreeol 忽略行尾差異。

^ignorewhitespaces 忽略空格差異。

^ignoreeolwhitespaces 忽略行尾和空格差異。

^notignore 偵測行尾和空格差異。

設定 PLASTICEDITOR 環境變數，以指定要輸入

註解的編輯器。

範例：

cm ^shelve -^c="我的註解"

(擱置目前工作區中所有暫止的變更，包括

註解。)

cm ^shelve file1.txt "file 2.txt" -^commentsfile=commentshelve.txt

(擱置所選暫止的變更並套用

commentshelve.txt 檔案中的註解。)

cm ^shelve --^apply=^sh:3

(套用已儲存的擱置集。)

cm ^shelve --^delete=^sh:3

(移除已儲存的擱置集。)

cm ^status --^short --^changelist=暫止\_待\_檢閱 | cm ^shelve -

(擱置用戶端變更清單。

以上命令會列出名為 '暫止\_待\_檢閱' 變更清單中

的路徑，且此路徑清單會重新導向至

擱置命令的輸入。)

== CMD\_DESCRIPTION\_SHELVESET ==

可讓使用者管理擱置集。

== CMD\_USAGE\_SHELVESET ==

用法：

cm ^shelveset <命令> [選項]

命令：

^create | ^mk

^delete | ^rm

^apply

若要取得每個命令的詳細資訊，請執行：

cm ^shelveset <命令> --^usage

cm ^shelveset <命令> --^help

== CMD\_HELP\_SHELVESET ==

範例：

cm ^shelveset ^create -^c="我的註解"

cm ^shelveset ^delete ^sh:3

cm ^shelve ^apply ^sh:3

== CMD\_DESCRIPTION\_SHELVESET\_CREATE ==

擱置暫止的變更。

== CMD\_USAGE\_SHELVESET\_CREATE ==

用法：

cm ^shelveset ^create | ^mk [<項目\_路徑>[ ...]] [--^all] [--^dependencies]

[-^c=<字串\_註解> | -^commentsfile=<註解\_檔案>]

選項：

項目\_路徑 要擱置的項目。使用空格分隔使用者名稱。

使用雙引號 (" ") 指定含空格的

路徑。

--^all 指定路徑上在本地變更、移動和刪除的項目

也會包括在內。

--^dependencies 將本地變更相依性包含在

要擱置的項目中。

-^c 將指定註解套用至已建立的擱置。

-^commentsfile 將指定檔案中的註解套用至已建立的

擱置。

== CMD\_HELP\_SHELVESET\_CREATE ==

'^shelveset ^create' 命令會將已簽出項目的內容儲存在

儲存庫內。如此一來，內容就會受到保護，而不需要

簽入檔案。

備註：

如果未指定 <項目\_路徑> 或任何選項，擱置集將

包含工作區中所有暫止的變更。

'^shelveset ^create' 作業一律會從指定路徑以遞迴方式

套用。

擱置項目的要求：

- 項目必須受原始程式碼控制。

- 項目必須簽出或變更 (必須使用 '--^all' 選項)。

設定 PLASTICEDITOR 環境變數，以指定要輸入

註解的編輯器。

範例：

cm ^shelveset ^create -^c="我的註解"

(擱置目前工作區中所有暫止的變更，包括

註解。)

cm ^shelveset file1.txt "file 2.txt" -^commentsfile=commentshelve.txt

(擱置所選暫止的變更並套用 'commentshelve.txt' 檔案中

的註解。請注意，'^create' 為預設子命令。)

cm ^status --^short --^changelist=暫止\_待\_檢閱 | cm ^shelveset -

(擱置用戶端變更清單。

以上命令會列出名為 '暫止\_待\_檢閱' 變更清單中

的路徑，且此路徑清單會重新導向至

'^shelveset' 命令的輸入。)

== CMD\_DESCRIPTION\_SHELVESET\_DELETE ==

刪除擱置集。

== CMD\_USAGE\_SHELVESET\_DELETE ==

用法：

cm ^shelveset ^delete | ^rm <擱置\_規格>

擱置\_規格 擱置集規格。(使用 'cm ^help ^objectspec'

以深入瞭解擱置集規格。)

== CMD\_HELP\_SHELVESET\_DELETE ==

'^shelveset ^delete' 命令會刪除擱置集。

範例：

cm ^shelveset ^delete ^sh:3

(移除已儲存的擱置集。)

== CMD\_DESCRIPTION\_SHELVESET\_APPLY ==

套用已儲存的擱置集。

== CMD\_USAGE\_SHELVESET\_APPLY ==

用法：

cm ^shelveset ^apply <擱置\_規格> [--^mount] [--^encoding=<名稱>]

[--^comparisonmethod=(^ignoreeol | ^ignorewhitespaces| \

^ignoreeolwhitespaces | ^notignore)]

擱置\_規格 擱置集規格。(使用 'cm ^help ^objectspec'

以深入瞭解擱置集規格。)

選項：

--^mount 指定儲存庫的裝載點。

--^encoding 指定輸出編碼，例如：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

--^comparisonmethod 設定比較方法。如需詳細資訊，請參閱備註。

== CMD\_HELP\_SHELVESET\_APPLY ==

'^shelveset ^apply' 命令會還原已儲存擱置集的內容。

備註：

比較方法:

^ignoreeol 忽略行尾差異。

^ignorewhitespaces 忽略空格差異。

^ignoreeolwhitespaces 忽略行尾和空格差異。

^notignore 偵測行尾和空格差異。

範例：

cm ^shelveset ^apply ^sh:3

(套用已儲存擱置。)

== CMD\_DESCRIPTION\_SHOW\_FIND\_OBJECTS ==

列出物件和屬性。

== CMD\_USAGE\_SHOW\_FIND\_OBJECTS ==

用法：

cm ^showfindobjects

== CMD\_HELP\_SHOW\_FIND\_OBJECTS ==

可用物件與屬性：

^attribute:

您可使用下列欄位進行篩選以尋找屬性：

^type ：字串。

範例：

cm ^find ^attribute "^where ^type = '狀態'"

(尋找 '狀態' 類型的所有屬性。)

^value ：字串。

^date ：日期。

如需詳細資訊，請查看本指南中的「日期常數」。

範例：

cm ^find ^attribute "^where ^date > '^this ^week'"

(尋找本週已套用的所有屬性。)

^owner ：使用者。

准許特殊使用者 '^me'。

範例：

cm ^find ^attribute "^where ^value = '已解決' ^and ^owner = '^me'"

(尋找我已套用 '已解決' 值的所有屬性。)

^GUID ：全域唯一識別碼。

十六進位 ID (格式為 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)。

^comment ：字串。

^srcobj ：物件規格：項目路徑、分支、變更集、修訂或標籤。

使用 'cm ^help ^objectspec' 以瞭解如何指定這些物件。

範例：

cm ^find ^attribute "^where ^srcobj = '^item:readme.txt'"

(尋找已套用至項目 'readme.txt' 的屬性。)

cm ^find ^attribute "^where ^srcobj = '^br:/main/scm23343'"

(尋找已套用至 scm23343 分支的屬性。)

cm ^find ^attribute "^where ^srcobj = '^rev:readme.txt#^br:/main/task002'"

(尋找已套用至指定修訂版本的屬性。)

cm ^find ^attribute "^where ^srcobj = '^rev:^revid:1126'"

(尋找已套用至指定修訂 ID 的屬性。)

^ID ：整數。

^attributetype:

您可使用下列欄位進行篩選以尋找屬性類型：

^name ：字串。

範例：

cm ^find ^attributetype "^where ^name ^like 'st%'"

(尋找名稱開頭為 'st' 的所有屬性。)

^value ：字串。

^date ：日期。

如需詳細資訊，請查看本指南中的「日期常數」。

範例：

cm ^find ^attribute "^where ^date > '^today'"

(尋找今天已套用的所有屬性。)

^owner ：使用者。

准許特殊使用者 '^me'。

^GUID ：全域唯一識別碼。

十六進位 ID (格式為 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)。

^comment ：字串。

範例：

cm ^find ^attributetype "^where ^comment != ''" --^xml

(尋找具有註解的所有屬性類型，並

以 XML 格式將輸出列印至標準輸出。)

^source ：物件規格：項目路徑、分支、變更集或標籤。

使用 'cm ^help ^objectspec' 以瞭解如何指定這些物件。

範例：

cm ^find ^attributetype "^where ^source = '^item:readme.txt'"

(尋找項目 'readme.txt' 中的所有屬性類型。)

cm ^find ^attributetype "^where ^source = '^cs:30'"

(尋找變更集 '30' 中的所有屬性類型。)

cm ^find ^attributetype "^where ^source = '^lb:v0.14.1'"

(尋找標籤 'v0.14.1' 中的所有屬性類型。)

^ID ：整數。

複寫欄位。檢查下列「複寫相關欄位」。

^ReplLogId

^ReplSrcDate

^ReplSrcId

^ReplSrcRepository

^ReplSrcServer

^branch:

您可使用下列欄位進行篩選以尋找分支：

^name ：字串。

範例：

cm ^find ^branch "^where ^name ^like 'scm23%'"

(尋找名稱開頭為 'scm23' 的分支。)

^date ：日期。

如需詳細資訊，請查看本指南中的「日期常數」。

範例：

cm ^find ^branch "^where ^date > '^one ^week ^ago'"

(尋找上週已建立的分支。)

^changesets ：日期 (分支中變更集的日期)。

如需詳細資訊，請查看本指南中的「日期常數」。

範例：

cm ^find ^branch "^where ^changesets >= '^today'"

(尋找今天已建立且含有變更集的分支。)

^attribute ：字串。

^attrvalue ：字串。

範例：

cm ^find ^branch "^where ^attribute = '狀態' ^and ^attrvalue = '失敗'"

(尋找具有 '狀態' 屬性且

值為 '失敗' 的分支。)

^owner ：使用者。

准許特殊使用者 '^me'。

^parent ：分支規格。

使用 'cm ^help ^objectspec' 以瞭解如何指定此物件。

範例：

cm ^find ^branch "^where ^owner != '^me' ^and ^parent != '^br:/main'"

(尋找由其他人建立且其上層

分支並非 '/main' 的分支。)

^comment ：字串。

^GUID ：全域唯一識別碼。

十六進位 ID (格式為 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)。

^ID ：整數。

範例：

cm ^find ^branch "^where ^id = 2029607"

(尋找 ID 為 2029607 的分支。)

複寫欄位。檢查下列「複寫相關欄位」。

^ReplLogId

^ReplSrcDate

^ReplSrcId

^ReplSrcRepository

^ReplSrcServer

^changeset:

您可使用下列欄位進行篩選以尋找變更集：

^branch ：分支規格。

使用 'cm ^help ^objectspec' 以瞭解如何指定此

物件。

範例：

cm ^find ^changeset "^where ^branch = '/main/scm23119'"

(尋找分支 'scm23119' 中的所有變更集。)

^changesetid ：整數。

^attribute ：字串。

範例：

cm ^find ^changeset "^where ^attribute = '狀態'"

(尋找具有 '狀態' 屬性的變更集。)

^attrvalue ：字串。

^date ：日期。

如需詳細資訊，請查看本指南中的「日期常數」。

^owner ：使用者。

准許特殊使用者 '^me'。

範例：

cm ^find ^changeset "^where ^date >= '2018/8/6' ^and ^owner != '^me'"

(尋找建立日期大於或等於

2018/8/6 且由其他人建立的所有變更集。)

^GUID ：全域唯一識別碼。

十六進位 ID (格式為 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)。

範例：

cm ^find ^changeset "^where ^guid = '1b30674f-14cc-4fd7-962b-676c8a6f5cb6'"

(尋找具有指定 GUID 的變更集。)

^comment ：字串。

範例：

cm ^find ^changeset "^where ^comment = ''"

(尋找不含任何註解的變更集。)

^onlywithrevisions：布林值。

篩選變更集是否有修訂。

範例：

cm ^find ^changeset "^where ^onlywithrevisions = 'false'"

(尋找不含任何修訂的變更集。)

^returnparent ：布林值。

回到變更集上層的方式。適合用於編寫指令碼。

範例：

cm ^find ^changeset "^where ^changesetid = 29 ^and ^returnparent = 'true'"

(尋找變更集 29 的上層。)

^parent ：變更集 ID (整數)。

範例：

cm ^find ^changeset "^where ^parent = 548"

(尋找上層為變更集 548 的所有變更集。)

^ID ：整數。

複寫欄位。檢查下列「複寫相關欄位」。

^ReplLogId

^ReplSrcDate

^ReplSrcId

^ReplSrcRepository

^ReplSrcServer

^label:

您可使用下列欄位進行篩選以尋找標籤：

^name ：字串。

範例：

cm ^find ^label "^where ^name ^like '7.0.16.%'"

(尋找名稱開頭為 '7.0.16' 的標籤。)

^attribute ：字串。

^attrvalue ：字串。

^date ：日期。

如需詳細資訊，請查看本指南中的「日期常數」。

範例：

cm ^find ^label "^where ^date >= '^this ^month' ^and \

^attribute = '發佈-狀態' ^and ^attrvalue != '已發佈'"

(尋找本月已建立且 '發佈-狀態' 屬性

設為 '已發佈' 以外的值的標籤。)

^owner ：使用者。

准許特殊使用者 '^me'。

^GUID ：全域唯一識別碼。

十六進位 ID (格式為 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)。

^branch ：分支規格。

使用 'cm ^help ^objectspec' 以瞭解如何指定此物件。

範例：

cm ^find ^label "^where ^branch = '/main'"

(尋找已套用至主要分支的所有標籤。)

^branchid ：整數。

^changeset ：變更集 ID (整數)。

範例：

cm ^find ^label "^where ^changeset = 111733"

(尋找已套用至變更集 111733 的標籤。)

^comment ：字串。

^ID ：整數。

複寫欄位。檢查下列「複寫相關欄位」。

^ReplLogId

^ReplSrcDate

^ReplSrcId

^ReplSrcRepository

^ReplSrcServer

^merge:

您可使用下列欄位進行篩選以尋找合併：

^srcbranch ：分支規格。

使用 'cm ^help ^objectspec' 以瞭解如何指定此物件。

範例：

cm ^find ^merge "^where ^srcbranch = '^br:/main'"

(尋找來自主要分支的合併。)

^srcchangeset ：變更集 ID (整數)。

^dstbranch ：分支規格。

使用 'cm ^help ^objectspec' 以瞭解如何指定此物件。

^dstchangeset ：變更集 ID (整數)。

範例：

cm ^find ^merge "^where ^dstchangeset = 108261" \

--^format="{^srcbranch} {^srcchangeset} {^dstbranch} {^dstchangeset} {^owner}"

(尋找變更集 108261 的合併，並列印

顯示來源 (分支和變更集 ID)、

目的地 (分支和變更集 ID) 以及合併擁有者的格式化輸出。)

^date ：日期。

如需詳細資訊，請查看本指南中的「日期常數」。

^owner ：使用者。

准許特殊使用者 '^me'。

^GUID ：全域唯一識別碼。

十六進位 ID (格式為 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)。

^type ：字串。

可能的值為 '^merge'、'^cherrypick'、

'^cherrypicksubstractive'、'^interval'、'^intervalcherrypick'

和 '^intervalcherrypicksubstractive'

範例：

cm ^find ^merge "^where ^type = '^cherrypick' ^and ^owner = '^me'"

(尋找我所有精心挑選的項目。)

^ID ：整數。

^replicationlog:

您可使用下列欄位進行篩選以尋找複寫記錄：

^branch ：分支規格。

使用 'cm ^help ^objectspec' 以瞭解如何指定此物件。

範例：

cm ^find ^replicationlog "^where ^branch = '/main/gm22358'"

(尋找分支 'gm22358' 的複寫記錄。)

^repositoryname ：字串。

^owner ：使用者。

准許特殊使用者 '^me'。

^date ：日期。

如需詳細資訊，請查看本指南中的「日期常數」。

^server ：字串。

^package ：布林值。

範例：

cm ^find ^replicationlog "^where ^package = 'T' ^and ^server ^like '%cloud%'"

(尋找從伺服器名稱包含 'cloud' 之套件

所建立的複寫記錄。)

^ID ：整數。

^review:

您可使用下列欄位進行篩選以尋找程式碼檢閱：

^status ：字串。

^assignee ：字串。

範例：

cm ^find ^review "^where ^status = '暫止' ^and ^assignee = '^me'"

(尋找我所有暫止的檢閱。)

^title ：字串。

^target ：物件規格：分支或變更集。

使用 'cm ^help ^objectspec' 以瞭解如何指定此物件。

範例：

cm ^find ^review "^where ^target = '^br:/main/scm17932'"

(尋找分支 'scm17932' 的相關檢閱。)

^targetid ：整數。

^targettype ：字串。

可能的值為 '^branch' 和 '^changeset'。

範例：

cm ^find ^review "^where ^targettype = '^changeset'"

(尋找目標類型為變更集的檢閱。)

^date ：日期。

如需詳細資訊，請查看本指南中的「日期常數」。

^owner ：使用者。

准許特殊使用者 '^me'。

^GUID ：全域唯一識別碼。

十六進位 ID (格式為 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)。

^ID ：整數。

^revision:

您可使用下列欄位進行篩選以尋找修訂：

^branch ：分支規格。

使用 'cm ^help ^objectspec' 以瞭解如何指定此物件。

^changeset ：變更集 ID (整數)。

範例：

cm ^find ^revision "^where ^changeset >= 111756"

(尋找變更集 111756

和更新版本中建立的修訂。)

^item ：字串或整數。

^itemid ：整數。

範例：

cm ^find ^revision "^where ^item = 'readme.txt' ^or ^itemid = 2250"

(尋找項目 'readme.txt' 的修訂以及

項目 ID 2250。)

cm ^find ^revision "^where ^item = 'readme.txt' ^or ^item = 2250"

(取得與先前範例相同的修訂。)

^attribute ：字串。

^attrvalue ：字串。

範例：

cm ^find ^revision "^where ^attribute = '狀態' ^and ^attrvalue != '未解決'"

(尋找具有 '狀態' 屬性且值

並非 '未解決' 的修訂。)

^archived ：布林值。

範例：

cm ^find ^revision "^where ^archived = 'true'"

(尋找外部儲存空間中的

已封存修訂。)

^comment ：字串。

^date ：日期。

如需詳細資訊，請查看本指南中的「日期常數」。

^GUID ：全域唯一識別碼。

十六進位 ID (格式為 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)。

^owner ：使用者。

准許特殊使用者 '^me'。

^parent ：修訂 ID (整數)。

^returnparent ：布林值。

^shelve ：擱置 ID (整數)。

^size ：整數 (以位元組為單位)。

^type ：字串。

可能的值為 '^dir'、'^bin' 和 '^txt'。

範例：

cm ^find ^revision "^where ^type = '^txt' 和 \

^size > 300000 ^and ^owner = '^me' and ^date >= '2 ^months ^ago'"

(尋找我在兩個月之前建立

且大小約大於 3MB 的文字修訂。)

^workspacecheckoutid ：整數。

^ID ：整數。

複寫欄位。檢查下列「複寫相關欄位」。

^ReplLogId

^ReplSrcDate

^ReplSrcId

^ReplSrcRepository

^ReplSrcServer

^shelve:

您可使用下列欄位進行篩選以尋找擱置：

^owner ：使用者。

准許特殊使用者 '^me'。

^date ：日期。

如需詳細資訊，請查看本指南中的「日期常數」。

範例：

cm ^find ^shelve "^where ^owner != '^me' ^and ^date >= '^1 ^years ^ago'"

(尋找其他人在去年

期間建立的擱置。)

^attribute ：字串。

^attrvalue ：字串。

^comment ：字串。

^GUID ：全域唯一識別碼。

十六進位 ID (格式為 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)。

^parent ：整數。

^shelveid ：整數。

範例：

cm ^find ^shelve "^where ^shelveid = 2"

(尋找具有名稱 2 的擱置。)

^ID ：整數。

範例：

cm ^find ^shelve "^where ^id >= 3848"

(尋找具有 ID 大於 3848 之物件的擱置。)

複寫欄位。檢查下列「複寫相關欄位」。

^ReplLogId

^ReplSrcDate

^ReplSrcId

^ReplSrcRepository

^ReplSrcServer

複寫相關欄位：

很多物件會追蹤複寫資料，這表示 Plastic 會追蹤

原本建立這些資料的所在位置。

您可以使用的欄位包括：

^ReplSrcServer ：儲存庫規格。代表「複寫來源伺服器」。

已從中複寫物件的伺服器。

範例：

cm ^find ^branch "^where ^replsrcserver='skull.codicefactory.com:9095'"

(尋找從伺服器 'skull' 複寫的分支。)

^ReplSrcRepository ：字串。代表「複寫來源儲存庫」。這是

已從中複寫物件的儲存庫。

範例：

cm ^find ^branch "^where ^replsrcserver = 'skull.codicefactory.com:9095' \

^and ^replsrcrepository = 'codice'"

(尋找已從伺服器 'skull' 和儲存庫

'codice' 複寫的分支。)

^ReplLogId ：整數。複寫作業的 ID。在 Plastic 中，

每次從複本建立新物件時，

都會建立新的「複寫記錄」。

範例：

cm ^find ^revision "^where ^repllogid = 2019974"

(尋找從複本

2019974 複寫的修訂。)

^ReplSrcDate ：日期。這是複本實際發生的

日期。

複寫物件會保留其原始建立日期，

而如果您想要尋找在特定時間

範圍內複寫的物件，

此欄位會非常有用。

範例：

cm ^find ^label "^where ^replsrcdate >= '^one ^month ^ago' \

^and ^date >= '15 ^days ^ago'"

(尋找在 15 天之前建立且在一個月

之前複寫的標籤。)

cm ^find ^replicationlog "^where ^date > '^one ^week ^ago'"

8780433 2018/09/27 8:49:38 codice@BACKYARD:8087 F mbarriosc

(尋找一週之前建立的複寫記錄。)

現在，您可以檢查是否已在複寫前

建立複寫分支：

cm ^find ^branch "^where ^repllogid = 8780433"

8780443 26/09/2018 12:20:55 /main/scm23078 maria codice T

^ReplSrcId ：整數。這是複寫來源伺服器的 ID。

您可使用 'cm ^find' 命令搜尋

'^replicationsource' 物件以探索此 ID。

範例：

cm ^find ^replicationsource

7860739 codice@AFRODITA:8087 d9c4372a-dc55-4fdc-ad3d-baeb2e975f27

8175854 codice@BACKYARD:8087 66700d3a-036b-4b9a-a26f-adfc336b14f9

現在，您可從以下尋找複寫的變更集：

codice@AFRODITA:8087:

cm ^find ^changesets "^where ^replsrcid = 7860739"

日期常數：

您可以使用遵循您的機器當地語系化設定的日期格式。

例如，如果您的電腦顯示 'MM-dd-yyyy' 格式的日期，

，則您可以在查詢中使用 '12-31-2019' 等日期。

您也可以使用下列常數簡化查詢：

'^today' ：今天的日期。

'^yesterday' ：昨天的日期。

'^this ^week' ：本週星期一的日期。

'^this ^month' ：本月第 1 天的日期。

'^this ^year' ：今年 1 月 1 日的日期。

'^one ^day ^ago' ：目前日期的一天前。

'^one ^week ^ago' ：目前日期的七天前。

'^one ^month ^ago' ：目前日期的一個月前。

'n ^days ^ago' ：目前日期的「n」天前。

'n ^months ^ago' ：目前日期的「n」個月前。

'n ^years ^ago' ：目前日期的「n」年前。

下列 '^where' 子句適用於欄位類型 '^date':

'(...)^where ^date > '^today' (...)'

'(...)^where ^date < '^yesterday' (...)'

'(...)^where ^date > '^this ^week' (...)'

'(...)^where ^date > '^this ^month' (...)'

'(...)^where ^date < '^one ^day ^ago' ^and ^date > '3 ^days ^ago' (...)'

'(...)^where ^date < '^one ^week ^ago' ^and ^date > '3 ^weeks ^ago' (...)'

'(...)^where ^date < '^one ^month ^ago' ^and ^date > '3 ^months ^ago' (...)'

'(...)^where ^date > '1 ^year ^ago' (...)'

您也可以在 'cm ^find' 命令上透過

--^dateformat 標誌強制使用特定日期格式。請查看 'cm ^find --^help' 以瞭解詳細資訊。

== CMD\_DESCRIPTION\_TRIGGER\_SHOWTYPES ==

顯示可用的觸發類型。

== CMD\_USAGE\_TRIGGER\_SHOWTYPES ==

用法：

cm ^trigger ^showtypes

== CMD\_DESCRIPTION\_SHOWACL ==

顯示物件的 ACL。

== CMD\_USAGE\_SHOWACL ==

用法：

cm ^showacl | ^sa <物件\_規格> [--^extended] [--^xml[=<輸出\_檔案>]]

[--^encoding=<名稱>]

物件\_規格 要顯示 ACL 的物件規格。

此命令的有效物件為：

儲存庫伺服器、儲存庫、分支、變更集、標籤、項目，

以及屬性。

(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

選項：

--^extended 顯示 ACL 階層樹。

--^xml 將 XML 格式的輸出列印至標準輸出。

可以指定輸出檔案。

--^encoding 與 '--^xml' 選項搭配使用，指定要在

XML 輸出中使用的編碼，亦即：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

== CMD\_HELP\_SHOWACL ==

範例：

cm ^showacl ^repserver:PlasticServer:8084

(顯示所選伺服器的 ACL。)

cm ^sa ^br:/main --^extended

(顯示所選分支規格的 ACL 階層樹。)

== CMD\_DESCRIPTION\_SHOWCOMMANDS ==

顯示所有可用命令。

== CMD\_USAGE\_SHOWCOMMANDS ==

用法：

cm ^showcommands

== CMD\_HELP\_SHOWCOMMANDS ==

== CMD\_DESCRIPTION\_SHOWOWNER ==

顯示物件的擁有者。

== CMD\_USAGE\_SHOWOWNER ==

用法：

cm ^showowner | ^so <物件\_規格>

物件\_規格 要顯示擁有者的物件規格。

物件必須是下列其中一種：

儲存庫伺服器、儲存庫、分支、變更集、

標籤、屬性、修訂和項目。

(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

== CMD\_HELP\_SHOWOWNER ==

備註：

此命令會顯示物件的擁有者。擁有者可為使用者或

群組。可使用 'cm ^setowner' 命令修改擁有者。

範例：

cm ^showowner ^repserver:PlasticServer:8084

(顯示所選伺服器的擁有者。)

cm ^so ^item:samples\

(顯示所選項目規格的擁有者。)

== CMD\_DESCRIPTION\_SHOWPERMISSIONS ==

列出可用權限。

== CMD\_USAGE\_SHOWPERMISSIONS ==

用法：

cm ^showpermissions | ^sp

== CMD\_HELP\_SHOWPERMISSIONS ==

範例：

cm ^showpermissions

== CMD\_DESCRIPTION\_SHOWSELECTOR ==

顯示工作區選擇器。

== CMD\_USAGE\_SHOWSELECTOR ==

此命令已棄用。它的存在是為了回溯相容性，

但在 Plastic SCM 4.0 中大多已棄用選擇器。選擇器仍

存在是為了指定工作分支或變更集，但舊規則

篩選路徑的功能已不再受到支援。

用法：

cm ^showselector | ^ss [<工作區\_路徑> | <工作區\_規格>]

工作區\_路徑 要顯示選擇器的工作區路徑。

工作區\_規格 工作區規格。(使用 'cm ^help ^objectspec' 以

深入瞭解工作區規格。)

== CMD\_HELP\_SHOWSELECTOR ==

備註：

如果未指定路徑或工作區規格，命令將以

目前目錄做為工作區路徑。

範例：

cm ^showselector c:\workspace

(顯示所選工作區路徑的選擇器。)

cm ^ss

(顯示目前工作區的選擇器。)

cm ^showselector > mySelector.txt

(將目前工作區的選擇器寫入檔案。)

cm ^showselector ^wk:mywk@reptest

(顯示 'reptest' 儲存庫中 'mywk' 工作區的選擇器。)

== CMD\_DESCRIPTION\_SUPPORT ==

可讓使用者執行支援相關作業。

== CMD\_USAGE\_SUPPORT ==

用法：

cm ^support <命令> [選項]

命令：

^bundle

若要取得每個命令的詳細資訊，請執行：

cm ^support <命令> --^usage

cm ^support <命令> --^help

== CMD\_HELP\_SUPPORT ==

範例：

cm ^support

cm ^support ^bundle

cm ^support ^bundle c:\outputfile.zip

== CMD\_DESCRIPTION\_SUPPORT\_BUNDLE ==

建立含有相關記錄的「支援組合」套件。

您可以在要求協助、索取額外資訊或

提交錯誤時附加檔案。

== CMD\_USAGE\_SUPPORT\_BUNDLE ==

用法：

cm ^support ^bundle [<輸出檔案>]

選項：

輸出檔案 在指定的位置

建立「支援組合」套件。

== CMD\_HELP\_SUPPORT\_BUNDLE ==

備註：

此命令可讓使用者建立「支援組合」套件，以便

在要求協助、索取額外資訊或提交錯誤時附加。

使用者可選擇是否要指定輸出檔案的位置；如果未指定，

輸出檔案將寫入暫存目錄。

範例：

cm ^support ^bundle

(在暫存目錄中建立「支援組合」。)

cm ^support ^bundle c:\outputfile.zip

(在指定的位置建立「支援組合」。)

== CMD\_DESCRIPTION\_SWITCH ==

將工作區切換至分支、變更集、標籤或擱置集。

== CMD\_USAGE\_SWITCH ==

用法：

cm ^switch (<分支規格> | <變更及規格> | <標籤規格> | <擱置集規格>)

[--^workspace=<路徑>] [--^repository=<名稱>]

[--^forcedetailedprogress]

(使用 'cm ^help ^objectspec' 以深入瞭解分支、變更集、標籤

和擱置集規格。)

選項：

--^workspace 工作區所在位置的路徑。

--^repository 切換至指定儲存庫。

--^forcedetailedprogress 強制顯示詳細進度，即使在標準輸出

重新導向的情況下亦然。

== CMD\_HELP\_SWITCH ==

備註：

此命令可讓使用者將工作區樹更新為指定物件

(分支、標籤、擱置集或變更集) 的內容。

範例：

cm ^switch ^br:/main

cm ^switch ^lb:Rel1.1

cm ^switch ^br:/main/scm002 --^repository=rep2

cm ^switch ^cs:4375

cm ^switch ^sh:2

== CMD\_DESCRIPTION\_SWITCH\_TO\_BRANCH ==

將分支設定為工作分支。

== CMD\_USAGE\_SWITCH\_TO\_BRANCH ==

此命令已棄用。請改用 cm 切換。

用法：

cm ^switchtobranch [選項] [分支\_規格]

分支\_規格：分支規格。

選項：

--^label=名稱 | --^changeset=數字：從指定的標籤或變更集

載入修訂。如果未指定分支\_規格，則需要

下列其中一個選項。

--^changeset=變更集：切換至指定的變更集。

--^repository=儲存庫：切換至指定的儲存庫。

--^workspace | -工作區=路徑：工作區所在位置的路徑。

== CMD\_HELP\_SWITCH\_TO\_BRANCH ==

備註：

此命令可讓使用者在分支中工作。

如果未指定分支\_規格，則必須指定標籤或變更集。

如果未指定儲存庫，分支則會設定為目前儲存庫。

範例：

cm ^switchtobranch ^br:/main

cm ^switchtobranch ^br:/main/task001

cm ^switchtobranch --^label=BL050

(唯讀組態。此命令會載入已標示

變更集的內容。)

== CMD\_DESCRIPTION\_SYNC ==

與 Git 同步。

== CMD\_USAGE\_SYNC ==

用法：

cm ^synchronize | ^sync <儲存庫規格> ^git [<url> [--^user=<使用者\_名稱> --^pwd=<密碼>]]

[(--^txtsimilaritypercent | --^binsimilaritypercent | \

--^dirsimilaritypercent)=<值>]

[--^author] [--^skipgitlfs]

儲存庫規格 儲存庫規格。(使用 'cm ^help ^objectspec'

以深入瞭解儲存庫規格。)

git (預設)。

選項：

url 遠端儲存庫 URL (http(s):// 或 git:// 或

SSH URL)。

--^user 指定 URL 的使用者名稱。

--^pwd 指定 URL 的密碼。

--^txtsimilaritypercent \

--^binsimilaritypercent \

--^dirsimilaritypercent

可偵測已移動的項目，與 Plastic SCM

GUI 的方式相同。

--^author 使用 git 作者的名稱和時間戳記值。

(預設為 git 提交者)

--^skipgitlfs 忽略

.gitattributes 檔案中的 Git LFS 組態。它的功能與沒有 Git LFS

支援的情況類似。

== CMD\_HELP\_SYNC ==

備註：

如果 git 儲存庫要求輸入使用者和密碼，則請使用 '^url'、'--^user'

和 '--^pwd' 選項。

如果 git 儲存庫未要求輸入使用者和密碼，則請將 '^url'

選項與第一個同步作業搭配使用。對於後續同步作業，'^url'

為選用選項。

若要使用 SSH 通訊協定執行同步作業，您必須將「ssh」用戶端

新增至 PATH 環境變數並正確地設定，才能連線至

遠端主機 (亦即已設定私密/公開金鑰)。

範例：

cm ^sync default@localhost:8087 ^git git://localhost/repository

== CMD\_DESCRIPTION\_TRIGGER ==

可讓使用者管理觸發器。

== CMD\_USAGE\_TRIGGER ==

用法：

cm ^trigger | ^tr <命令> [選項]

命令：

^create | ^mk

^delete | ^rm

^edit

^list | ^ls

^showtypes

若要取得每個命令的詳細資訊，請執行：

cm ^trigger <命令> --^usage

cm ^trigger <命令> --^help

== CMD\_HELP\_TRIGGER ==

範例：

cm ^tr ^mk ^before-mklabel new "/path/to/script" --^server=myserver:8084

cm ^tr ^edit ^before-mklabel 7 --^position=4 --^server=myserver:8084

cm ^tr ^ls ^before-mkbranch --^server=myserver:8084

cm ^tr ^rm ^after-setselector 4

cm ^tr ^showtypes

== CMD\_DESCRIPTION\_TUBE ==

執行 Plastic Tube 的相關命令。

== CMD\_USAGE\_TUBE ==

用法：

cm ^tube ^config -^u=<使用者> -^p=<密碼>

(將 Plastic SCM 設定為使用 Plastic Tube 搭配指定的使用者和

密碼。)

cm ^tube ^create <遠端使用者>

(建立 "遠端使用者 -> 我的使用者" Tube。

"我的使用者" 使用者可讓 "遠端使用者" 連線至 "我的使用者" 伺服器。

可建立從 "遠端使用者" 到 "我的使用者" 的連線。

只能建立從其他使用者到目前 Tube 使用者的 Tube。)

cm ^tube ^remove <遠端使用者>

(移除 "遠端使用者 -> 我的使用者" Tube。)

cm ^tube ^local

(列出本地伺服器中共用的本地儲存庫，以及共用的

使用者。)

cm ^tube ^remote

(列出與目前 Tube 使用者共用的共用

遠端儲存庫。)

cm ^tube ^share <儲存庫\_規格>[ ...]-^u=<遠端使用者> -^a=(^pull | ^push | ^pull,^push)

(與遠端使用者共用本地儲存庫並設定指定的

存取模式。使用空格分隔儲存庫規格。)

cm ^tube ^unshare <儲存庫\_規格>[ ...]-^u=<遠端使用者>

(取消與遠端使用者共用本地儲存庫。使用空格

分隔儲存庫規格。)

cm ^tube ^connect

(將 Plastic SCM 伺服器連線至 Plastic Tube。)

cm ^tube ^disconnect

(中斷 Plastic SCM 伺服器與 Plastic Tube 的連線。)

cm ^tube ^status

(顯示 Plastic SCM 伺服器是否已連線至 Plastic Tube。)

選項

-^u Tube 使用者 (plasticscm.com 使用者)。

-^p 使用者密碼。

-^a 設定存取模式。

儲存庫\_規格 儲存庫規格。(使用 'cm ^help ^objectspec' 以

深入瞭解儲存庫規格。)

== CMD\_HELP\_TUBE ==

備註：

使用 'cm ^tube' 命令管理 Plastic Tube。

範例：

cm ^tube ^config -^u=ruben@codicesoftware.com -^p=rubenpassword

cm ^tube ^create pablo@codicesoftware.com

('pablo@codicesoftware.com' 可連線至目前的 Plastic Tube 使用者)

cm ^tube ^remove pablo@codicesoftware.com

cm ^tube ^local

cm ^tube ^remote

cm ^tube ^share repo@server:8087 -^u=pablo@codicesoftware.com -^a=^pull,^push

cm ^tube ^share repo@server:8087 doc@server:8087 -^u=pablo@codicesoftware.com -^a=^push

cm ^tube ^unshare repo@server:8087 -^u=pablo@codicesoftware.com

cm ^tube ^connect

cm ^tube ^disconnect

cm ^tube ^status

== CMD\_DESCRIPTION\_UNCO ==

復原項目簽出。

== CMD\_USAGE\_UNCO ==

用法：

cm ^undocheckout | ^unco <項目\_路徑>[ ...][-^a | --^all] [--^symlink] [--^silent]

[--^machinereadable [--^startlineseparator=<分隔符號>]

[--^endlineseparator=<分隔符號>] [--^fieldseparator=<分隔符號>]]

項目\_路徑 要套用作業的項目。使用空格分隔

路徑。使用雙引號 (" ") 指定含空格的

路徑。

使用 . 將作業套用至目前目錄。

選項：

-^a | --^all 復原指定項目中的所有變更。如果

項目已簽出，則會還原

簽出。如果項目已在本地修改，則會

還原修改。

--^symlink 將復原簽出作業套用至連結，而非

套用至目標。

--^silent 不顯示任何輸出。

--^machinereadable 以易於剖析的格式輸出結果。

--^startlineseparator 搭配 '--^machinereadable' 標誌使用，

指定各行應如何起始。

--^endlineseparator 搭配 '--^machinereadable' 標誌使用，

指定各行應如何結束。

--^fieldseparator 搭配 '--^machinereadable' 標誌使用，

指定應如何分隔欄位。

== CMD\_HELP\_UNCO ==

備註：

如果項目已簽出，而且您不想將其簽入，您可以使用此命令復原

簽出。檔案和資料夾皆可取消

簽出。項目將更新為簽出前的狀態。

要求：

- 項目必須受原始程式碼控制。

- 必須簽出項目。

從 stdin 讀取輸入：

'^undocheckout' 命令可從 stdin 讀取路徑。若要執行此操作，可

傳遞單一破折號「-」來完成。

範例：cm ^undocheckout ^checkin -

將讀取路徑，直到輸入空白行為止。

這可讓您使用管線指定要復原簽出的檔案。

範例：

dir /S /B \*.c | cm ^undocheckout --^all -

(在 Windows 中，復原工作區中所有 .c 檔案的簽出。)

範例：

cm ^undocheckout .

(復原目前目錄中的簽出。)

cm ^undocheckout file1.txt file2.txt

cm unco c:\workspace\file.txt

(復原所選檔案的簽出。)

cm ^unco -^a file1.txt

(復原 'file1.txt' 的簽出或本地修改)

cm ^unco 連結 --^symlink

(將復原簽出作業套用至連結檔案，而不是套用至目標；

此功能適用於 UNIX 環境。)

cm ^status --^short --^changelist=暫止\_待\_檢閱 | cm ^undocheckout -

(復原用戶端變更清單。

以上命令會列出名為

'暫止\_待\_檢閱' 變更清單中的路徑，且此路徑清單會重新導向至

復原簽出命令的輸入。)

cm ^unco .--^machinereadable

(復原目前目錄中的簽出，並以

簡化、易於剖析的格式列印結果。)

cm ^unco .--^machinereadable --^startlineseparator=">" --^endlineseparator="<" \

--^fieldseparator=","

(復原目前目錄中的簽出，並以簡化且易於剖析的格式

列印結果、起始行和結尾行，以及使用

指定的字串來分隔欄位。)

== CMD\_DESCRIPTION\_UNCOUNCHANGED ==

復原未變更的已簽出項目。

== CMD\_USAGE\_UNCOUNCHANGED ==

用法：

cm ^uncounchanged | ^unuc <項目\_路徑>[ ...][-^R | -^r | --^recursive]

[--^symlink] [--^silent]

項目\_路徑 要套用作業的項目。使用空格分隔

路徑。使用雙引號 (" ") 指定含空格的

路徑。

使用 . 將作業套用至目前目錄。

選項：

-^R 以遞迴方式復原指定路徑中未變更的項目。

--^symlink 將復原簽出未變更的作業套用至連結，而不是

套用至目標。

--^silent 不顯示任何輸出。

== CMD\_HELP\_UNCOUNCHANGED ==

備註：

此命令會從工作區根目錄以遞迴的方式套用。

從 stdin 讀取輸入：

'^uncounchanged' 命令可從 stdin 讀取路徑。若要執行此操作，可

傳遞單一破折號「-」來完成。

範例：cm ^uncounchanged -

將讀取路徑，直到輸入空白行為止。

這可讓您使用管線指定要復原簽出的未變更

檔案。

範例：

dir /S /B \*.c | cm ^uncounchanged -

(在 Windows 中，復原工作區中所有未變更 .c 檔案上的

簽出。)

範例：

cm ^uncounchanged .-^R

(在目前目錄上以遞迴方式復原未變更檔案的簽出。)

cm ^unuc /首頁/我的使用者/我的工作區/專案/範本 -^R

(在所選目錄上以遞迴方式復原未變更檔案的簽出。)

== CMD\_DESCRIPTION\_UNDELETE ==

取消刪除使用特定修訂的項目。

== CMD\_USAGE\_UNDELETE ==

用法：

cm ^undelete <修訂規格> <路徑>

修訂規格 將在工作區中載入之內容的

修訂的規格。(使用 'cm ^help ^objectspec'

以深入瞭解修訂規格。)

路徑 還原路徑。

== CMD\_HELP\_UNDELETE ==

備註：

要取消刪除的項目不應已載入工作區中。

xlink 不支援 '^undelete' 作業。

範例：

cm ^undelete ^revid:756 C:\mywks\src\foo.c

cm ^undelete ^itemid:68#^cs:2 C:\mywks\dir\myfile.pdf

cm ^undelete ^serverpath:/src#^br:/main C:\mywks\Dir

== CMD\_DESCRIPTION\_UNDOCHANGE ==

復原路徑上的變更。

== CMD\_USAGE\_UNDOCHANGE ==

用法：

cm ^undochange | ^unc <項目\_路徑>[ ...][-^R | -^r | --^recursive]

項目\_路徑 要套用作業的項目。使用空格分隔

路徑。使用雙引號 (" ") 指定含空格的

路徑。

使用 . 將作業套用至目前目錄。

選項：

-^R 以遞迴的方式套用作業。

== CMD\_HELP\_UNDOCHANGE ==

備註：

如果項目已簽出或修改，但未簽入，而且您不想

將其簽入，您可以使用此命令復原變更。項目

將更新為之前的內容。

從 stdin 讀取輸入：

'^undochange' 命令可從 stdin 讀取路徑。若要執行此操作，可

傳遞單一破折號「-」來完成。

範例：cm ^undochange -

將讀取路徑，直到輸入空白行為止。

這可讓您使用管線指定要復原變更的檔案。

範例：

dir /S /B \*.c | cm ^undochange -

(在 Windows 中，復原工作區中所有 .c 檔案的變更。)

範例：

cm ^unc .

(復原目前目錄上的檔案變更。)

cm ^undochange .-^R

(以遞迴方式復原目前目錄上的檔案變更。)

cm ^unc file1.txt "file 2.txt"

(復原所選檔案的變更。)

cm ^unc c:\workspace\file.txt

(復原所選檔案的變更。)

== CMD\_DESCRIPTION\_UNDO ==

復原工作區中的變更。

== CMD\_USAGE\_UNDO ==

用法：

cm ^undo [<路徑>[ ...]] [--^symlink] [-^r | --^recursive] [<篩選>[ ...]]

[--^silent | --^machinereadable [--^startlineseparator=<分隔符號>]

[--^endlineseparator=<分隔符號>] [--^fieldseparator=<分隔符號>]]

路徑 要套用作用的目標檔案或目錄的

路徑。使用雙引號 (" ") 指定含空格的

路徑。使用空格分隔路徑。

如果未指定路徑，復原作業預設會

使用目前目錄中的

所有檔案。

篩選 將指定的一或多個篩選套用至指定的

路徑。使用空格分隔篩選。如需詳細資訊，請參閱

＜篩選＞一節。

選項：

--^symlink 將復原作業套用至符號連結，而不是

套用至目標。

-^r 以遞迴方式執行復原。

--^silent 不顯示任何輸出。

--^machinereadable 以易於剖析的格式輸出結果。

--^startlineseparator 搭配 '--^machinereadable' 標誌使用，指定

各行應如何起始。

--^endlineseparator 搭配 '--^machinereadable' 標誌使用，指定

各行應如何結束。

--^fieldseparator 搭配 '--^machinereadable' 標誌使用，指定

應如何分隔欄位。

篩選：

如果未指定標誌，會依預設復原所有變更，但

可使用下列一或多個標誌篩選路徑。

如果檔案或目錄符合一或多個指定的變更類型，

將會復原所述之檔案或目錄上的所有變更。

例如，如果您指定 '--^checkedout' 和 '--^moved'，若檔案

已簽出且移動，則會復原這兩項變更。

--^checkedout 選取已簽出的檔案和目錄。

--^unchanged 選取未變更內容的檔案。

--^changed 選取已本地變更或簽出的檔案和

目錄。

--^deleted 選取已刪除的檔案和目錄。

--^moved 選取已移動的檔案和目錄。

--^added 選取已新增的檔案和目錄。

== CMD\_HELP\_UNDO ==

備註：

'^undo' 命令具有危險性 - 它會以無法回復的方式復原工作。

^undo 作業完成後，則無法復原受其影響的檔案和

目錄的先前狀態。如果未在引數中

指定路徑，預設將復原目前目錄中的所有變更，

但不是以遞迴方式進行。

這些等同於從 /src 目錄執行：

/src

|- file.txt

|- code.cs

\- /test

|- test\_a.py

\- test\_b.py

cm ^undo

cm ^undo \*

cm ^undo file.txt code.cs /test

cm ^undo .

cm ^undo /src file.txt code.cs

若要以遞迴的方式進行作業，您必須指定 '-^r' 標誌。

復原目錄下的所有變更 (包括影響到

目錄本身的變更)：

cm ^undo 目錄路徑 -^r

如果目錄路徑是工作區路徑，將會復原工作區中的

所有變更。

已刪除的項目：

若要復原檔案和目錄刪除作業，您必須指定項目的

完整路徑，或指定包含的目錄並使用遞迴 ('-^r')

標誌。

例如：

cm ^undo .

(不復原目前目錄中的刪除作業。)

cm ^undo .-^r

(以遞迴方式復原目前目錄中的所有刪除作業 (和其他變更)。)

cm ^undo src/file.txt

(復原 src/file.txt 的刪除作業 (或其他變更)。)

範例：

cm ^undo .-^r

(以遞迴方式復原目前目錄中的所有變更。如果已從

工作區的根目錄執行，則會復原整個工作區中的所有變更。)

cm ^co file.txt

cm ^undo file.txt

(在 'file.txt' 上復原簽出。)

^echo ^content >> file.txt

cm ^undo file.txt

(復原 'file.txt' 的本地變更。)

cm ^undo src

(復原 src 目錄及其檔案的變更。)

cm ^undo src/\*

(復原 src 中包含的所有檔案和目錄的變更，而

完全不影響 src。)

cm ^undo \*.cs

(復原符合目前目錄中 \*.cs 的

所有檔案或目錄的變更。)

cm ^undo \*.cs -^r

(復原符合目前目錄及其底下所有目錄中 \*.cs 的

所有檔案或目錄的變更。)

cm ^co file1.txt file2.txt

^echo ^content >> file1.txt

cm ^undo --^unchanged

(復原未變更 'file2.txt' 的簽出，進而忽略本地變更的

'file1.txt'。)

^echo ^content >> file1.txt

^echo ^content >> file2.txt

cm ^co file1.txt

cm ^undo --^checkedout

(復原已簽出檔案 'file1.txt' 中的變更，進而忽略 'file2.txt'，因為

這個檔案並未簽出。)

cm ^add file.txt

cm ^undo file.txt

(復原新增 'file.txt' 的動作，進而使其再次成為私人檔案。)

^rm file1.txt

^echo ^content >> file2.txt

cm ^add file3.txt

cm ^undo --^deleted --^added \*

(復原 'file1.txt' 刪除動作和 'file3.txt' 新增動作，而忽略 'file2.txt'

變更。)

== CMD\_DESCRIPTION\_LOCK\_UNLOCK ==

復原鎖定伺服器上的項目鎖定。

== CMD\_USAGE\_LOCK\_UNLOCK ==

用法：

cm ^lock ^unlock [<儲存庫伺服器規格>] <GUID>[ ...]

儲存庫伺服器規格 儲存庫伺服器規格。(使用 'cm ^help ^objectspec'

以深入瞭解儲存庫伺服器規格。)

GUID 要封鎖的 GUID 項目清單。使用空格

分隔 GUID。

== CMD\_HELP\_LOCK\_UNLOCK ==

備註：

- 此命令會使用指定的伺服器來解除鎖定項目。

- 如果未指定伺服器，此命令會嘗試從目前工作區

取得伺服器。

- 如果未在上述步驟中計算伺服器，則會從目前 Plastic SCM

用戶端組態取得伺服器。

- 只有伺服器管理員才能執行 'cm ^unlock' 命令。

- 若要指定 GUID，格式應為 32 位數並以連字號格式

分隔 (可選擇是否要加上括號)：

{00000000-0000-0000-0000-000000000000}

或 00000000-0000-0000-0000-000000000000

範例：

cm ^lock ^unlock 91961b14-3dfe-4062-8c4c-f33a81d201f5

(復原所選項目鎖定。)

cm ^lock ^unlock DIGITALIS:8084 2340b4fa-47aa-4d0e-bb00-0311af847865 \

bcb98a61-2f62-4309-9a26-e21a2685e075

(復原名為 'DIGITALIS' 的鎖定伺服器上所選的項目鎖定。)

cm ^lock ^unlock tardis@cloud 4740c4fa-56af-3dfe-de10-8711fa248635 \

71263c17-5eaf-5271-4d2c-a25f72e101d4

(復原名為 'tardis' 的雲端鎖定伺服器上所選的項目鎖定。)

== CMD\_DESCRIPTION\_UPDATE ==

更新工作區並下載最新變更。

== CMD\_USAGE\_UPDATE ==

用法：

cm ^update [<項目\_路徑> | --^last]

[--^changeset=<變更集規格>] [--^cloaked] [--^dontmerge] [--^forced]

[--^ignorechanges] [--^override] [--^recursewk] [--^skipchangedcheck]

[--^silent] [--^verbose] [--^xml[=<輸出\_檔案>]] [--^encoding=<名稱>]

[--^machinereadable [--^startlineseparator=<分隔符號>]

[--^endlineseparator=<分隔符號>] [--^fieldseparator=<分隔符號>]]

[--^forcedetailedprogress]

項目\_路徑 要更新的路徑。

使用 . 將更新套用至目前目錄。

如果未指定路徑，則會完全更新

目前工作區。

--^last 進行更新前，將工作區選擇器從變更集

組態或標籤組態變更為

分支組態。

選擇器已變更為變更集或

標籤所屬的分支。

選項：

--^changeset 將工作區更新為特定變更集。

(使用 'cm ^help ^objectspec' 以深入瞭解

變更集規格。)

--^cloaked 包含更新作業中的隱匿項目。

如果未指定此選項，將會在作業中忽略這些

隱匿的項目。

--^dontmerge 萬一更新作業期間需要更新合併，

則不會執行合併。

--^forced 將項目強制更新至選擇器中

指定的修訂。

--^ignorechanges 如果在更新工作區時偵測到

暫止的變更，則忽略顯示的

暫止變更警告訊息。

--^override 覆寫 Plastic SCM 控制範圍外的已變更檔案。

將以伺服器內容

覆寫其內容。

--^recursewk 更新目前路徑內找到的

所有工作區。可用來更新指定路徑中

包含的所有工作區。

--^skipchangedcheck 開始前，更新會先檢查工作區中是否有

本地變更。如果您總是在修改檔案前

簽出，您可以使用此檢查功能並

加快作業的速度。

--^silent 不顯示任何輸出。

--^verbose 顯示其他資訊。

--^xml 將 XML 格式的輸出列印至標準輸出。

可以指定輸出檔案。

--^encoding 與 --^xml 選項搭配使用，指定要在 XML 輸出中

使用的編碼，亦即：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

--^machinereadable 以易於剖析的格式輸出結果。

--^startlineseparator 搭配 '--^machinereadable' 標誌使用，

指定各行應如何起始。

--^endlineseparator 搭配 '--^machinereadable' 標誌使用，

指定各行應如何結束。

--^fieldseparator 搭配 '--^machinereadable' 標誌使用，

指定應如何分隔欄位。

--^forcedetailedprogress 強制顯示詳細進度，即使在標準輸出

重新導向的情況下亦然。

== CMD\_HELP\_UPDATE ==

備註：

'^update' 命令只會下載所需的檔案。

此命令應該會執行遞迴作業。

使用 '--^last' 選項時，不需要指定路徑。

在此情況下，將更新目前工作目錄所屬的

工作區。

(請切記，如果選擇器先前指向變更集或標籤，指定此標誌

可能會導致工作區選擇器變更為

分支組態。)

範例：

cm ^update

(更新目前工作區中的所有項目。)

cm ^update .

(更新目前目錄以及所有下層項目。)

cm ^update .--^forced --^verbose

(強制擷取所有修訂。)

cm ^update --^last

cm ^update .--^machinereadable --^startlineseparator=">"

(更新目前目錄並以簡化、

易於剖析的格式列印結果，使用指定的

字串起始行。)

== CMD\_DESCRIPTION\_VERSION ==

顯示目前用戶端版本號碼。

== CMD\_USAGE\_VERSION ==

用法：

cm ^version

== CMD\_HELP\_VERSION ==

== CMD\_DESCRIPTION\_WHOAMI ==

顯示目前的 Plastic SCM 使用者。

== CMD\_USAGE\_WHOAMI ==

用法：

cm ^whoami

== CMD\_HELP\_WHOAMI ==

== CMD\_USAGE\_WKTREENODESTATUS ==

用法：

cm ^wktreenodestatus 路徑 1, 路徑 2, ...

== CMD\_DESCRIPTION\_WORKSPACE ==

可讓使用者管理工作區。

== CMD\_USAGE\_WORKSPACE ==

用法：

cm ^workspace | ^wk <命令> [選項]

命令：

^list | ^ls

^create | ^mk

^delete | ^rm

^move | ^mv

^rename

若要取得每個命令的詳細資訊，請執行：

cm ^workspace <命令> --^usage

cm ^workspace <命令> --^help

== CMD\_HELP\_WORKSPACE ==

範例：

cm ^workspace ^create myWorkspace 工作區\_路徑

cm ^workspace ^list

cm ^workspace ^delete myWorkspace

== CMD\_DESCRIPTION\_WORKSPACE\_CREATE ==

建立新工作區。

== CMD\_USAGE\_WORKSPACE\_CREATE ==

用法：

cm ^workspace | ^wk [^create | ^mk] <工作區\_名稱> <工作區\_路徑> [<儲存庫\_規格>]

[--^selector[=<選擇器\_檔案>]

(建立新工作區。)

cm ^workspace | ^wk [^create | ^mk] <工作區\_名稱> <工作區\_路徑> --^dynamic --^tree=[<樹狀目錄>]

(建立動態工作區。此功能仍屬實驗性質，且

僅適用於 Windows。)

工作區\_名稱 新工作區名稱。

工作區\_路徑 新工作區的路徑。

儲存庫\_規格 使用指定的儲存庫建立新工作區。

儲存庫規格：查看 'cm ^help ^objectspec'。

選項：

--^selector 編輯新工作區的選擇器。

如果已指定選擇器檔案，則會從指定的檔案設定

新工作區的選擇器。

--^dynamic 建立動態工作區。此功能仍屬

實驗性質，且僅適用於 Windows。

指定此標誌需要使用 --^tree 參數。

--^tree 與 '--^dynamic' 標誌搭配使用，指定

動態工作區即將進行載入的初始點。這可為

分支、變更集或標籤規格。

工作區稍後將使用規格中的

儲存庫。(使用 'cm ^help ^objectspec' 以深入瞭解規格。)

== CMD\_HELP\_WORKSPACE\_CREATE ==

備註：

- 工作區是對應至本地檔案系統之儲存庫的檢視畫面。

工作區選擇器會定義指定工作區內容的規則。

使用 'cm ^showselector' 顯示工作區選擇器，或使用 'cm ^setselector'

進行修改。

- 如果未指定 '儲存庫\_規格' 或 '--^selector''，則會自動將工作區

設定為使用 client.conf 檔案中已設定之伺服器的第一個儲存庫

(依字母順序)。

- 動態工作區是實驗性質的功能 (僅限 Windows)，且

需要執行 plasticfs.exe 程式。

範例：

cm ^workspace ^create myworkspace c:\workspace

cm ^wk ^mk myworkspace /home/john/plastic\_view

(分別在 Windows 和 Linux 中建立 'myworkspace' 工作區。)

cm ^wk mywktest c:\wks\wktest --^selector=myselector.txt

(使用 'myselector.txt' 檔案中的選擇器，建立 'mywktest' 工作區。)

cm ^wk mywkprj c:\wks\wkprj myrep@^repserver:localhost:8084

(使用所選儲存庫建立 'mywkprj' 工作區。)

cm ^wk mywkprj c:\dynwks\mywkprj --^dynamic --^tree=^br:/main@myrep@localhost:8084

(使用 'myrep@localhost:8084'

儲存庫建立動態 'mywkprj' 工作區，在第一次裝載時指向 '^br:/main'。)

== CMD\_DESCRIPTION\_WORKSPACE\_DELETE ==

刪除工作區。

== CMD\_USAGE\_WORKSPACE\_DELETE ==

用法：

cm ^workspace | ^wk ^delete | ^rm [<工作區\_路徑> | <工作區規格>] [--^keepmetadata]

工作區\_路徑 要刪除的工作區路徑。

工作區規格 要刪除的工作區規格。(使用

'cm ^help ^objectspec' 以深入瞭解規格。)

選項:

--^keepmetadata 不會刪除 .plastic

資料夾中的中繼資料檔案。

== CMD\_HELP\_WORKSPACE\_DELETE ==

備註：

此命令會刪除路徑或規格所指定的工作區。

如果未指定引數，將直接使用目前工作區。

範例：

cm ^workspace ^delete

(移除目前工作區。)

cm ^wk ^delete c:\workspace

cm ^workspace rm /home/danipen/wks

cm ^wk ^rm ^wk:MiWorkspace

cm ^wk ^rm ^wk:MiWorkspace@DIGITALIS

== CMD\_DESCRIPTION\_WORKSPACE\_LIST ==

列出工作區。

== CMD\_USAGE\_WORKSPACE\_LIST ==

用法：

cm ^workspace | ^wk [^list | ^ls] [--^format=<字串\_格式>]

選項：

--^format 以指定的格式擷取輸出訊息。如需

詳細資訊，請參閱備註。

== CMD\_HELP\_WORKSPACE\_LIST ==

備註：

此命令接受格式字串以顯示輸出。

此命令的輸出參數如下所示：

{0} | {^wkname} 工作區名稱。

{1} | {^machine} 用戶端機器名稱。

{2} | {^path} 工作區路徑。

{3} | {^wkid} 工作區唯一識別碼。

{4} | {^wkspec} 使用此格式的工作區規格：

'wkname@machine'。

{^tab} 插入定位字元空格。

{^newline} 插入新行。

範例：

cm ^wk

(列出所有工作區。)

cm ^workspace ^list --^format={0}#{3,40}

cm ^workspace ^list --^format={^wkname}#{^wkid,40}

(列出所有工作區並顯示工作區名稱、# 符號和

40 個空格的工作區 GUID 欄位，靠左對齊。)

cm ^wk --^format="路徑 {2} 中的工作區 {0}"

cm ^wk --^format="路徑 {^path} 中的工作區 {^wkname}"

(列出所有工作區並以格式化字串的形式顯示結果。)

== CMD\_DESCRIPTION\_WORKSPACE\_MOVE ==

移動工作區。

== CMD\_USAGE\_WORKSPACE\_MOVE ==

用法：

cm ^workspace | ^wk ^move | ^mv [<工作區規格>] <新\_路徑>

選項：

工作區規格 要移動的工作區規格。(使用

'cm ^help ^objectspec' 以深入瞭解規格。)

新\_路徑 將移動至此處的工作區。

== CMD\_HELP\_WORKSPACE\_MOVE ==

備註：

此命令可讓使用者將工作區移動至磁碟上的另一個位置。

範例：

cm ^workspace ^move myWorkspace \new\workspaceDirectory

(將 'myWorkspace' 移動至指定的位置。)

cm ^wk ^mv c:\users\maria\wkspaces\newlocation

(將目前工作區移動至新位置。)

== CMD\_DESCRIPTION\_WORKSPACE\_RENAME ==

重新命名工作區。

== CMD\_USAGE\_WORKSPACE\_RENAME ==

用法：

cm ^workspace | ^wk ^rename [<工作區\_名稱>] <新\_名稱>

工作區\_名稱 要重新命名的工作區。

新\_名稱 工作區的新名稱。

== CMD\_HELP\_WORKSPACE\_RENAME ==

備註：

此命令會重新命名工作區。

如果未提供工作區名稱，則會使用目前工作區。

範例：

cm ^workspace ^rename mywk1 wk2

(將 'mywk1' 工作區重新命名為 'wk2'。)

cm ^wk ^rename 新名稱

(將目前工作區重新命名為 '新名稱'。)

== CMD\_DESCRIPTION\_WORKSPACESTATUS ==

顯示工作區中的變更。

== CMD\_USAGE\_WORKSPACESTATUS ==

用法：

cm ^status [<工作區\_路徑>] [--^changelist[=<名稱>] | --^changelists] [--^cutignored]

[ --^header] [ --^noheader] [ --^nomergesinfo] [ --^head]

[--^short] [--^symlink] [ --^dirwithchanges] [--^xml[=<輸出\_檔案>]]

[--^encoding=<名稱>] [ --^wrp | --^wkrootrelativepaths]

[--^fullpaths | --^fp] [<舊版\_選項>] [<搜尋\_類型>[ ...]]

[--^machinereadable [--^startlineseparator=分隔符號]

[--^endlineseparator=分隔符號] [--^fieldseparator=分隔符號]]

選項：

工作區\_路徑 將執行搜尋之工作區的

路徑。

--^changelist 顯示所選變更清單中的變更。

--^changelists 顯示用戶端變更清單中的分組變更。

--^cutignored 跳過已忽略目錄的內容。

需要 '--^ignored' 搜尋類型。如需詳細資訊，請參閱＜搜尋

類型＞一節。

--^header 只列印工作區狀態。

--^noheader 只列印已修改的項目搜尋結果。

--^nomergesinfo 不列印變更的合併資訊。

--^head 列印分支上最後一個變更集的狀態。

--^short 只列出包含變更的路徑。

--^symlink 將作業套用至符號連結，而不是套用至

目標。

--^dirwithchanges 顯示其中包含變更的目錄

(其中的已新增、已移動和已移除項目)。

--^xml 將 XML 格式的輸出列印至標準輸出。

可以指定輸出檔案。

--^encoding 與 --^xml 選項搭配使用，指定要在 XML 輸出中

使用的編碼，亦即：utf-8。

請參閱 MSDN 文件 (網址為

http://msdn.microsoft.com/en-us/library/system.text.encoding.aspx

)，以取得可支援之編碼及其格式的表格

(在頁尾的 [名稱] 欄中)。

--^wrp 列印工作區根目錄相對路徑，而非

目前目錄相對路徑。

--^fullpaths, --^fp 強制列印絕對路徑，覆寫任何其他

路徑列印設定。

--^machinereadable 以易於剖析的格式輸出結果。

--^startlineseparator 搭配 '--^machinereadable' 標誌使用，

指定各行應如何起始。

--^endlineseparator 搭配 '--^machinereadable' 標誌使用，

指定各行應如何結束。

--^fieldseparator 搭配 '--^machinereadable' 標誌使用，

指定應如何分隔欄位。

舊版選項：

--^cset 以舊版格式列印工作區狀態。

--^compact 以舊版格式列印工作區狀態和

變更清單。

--^noheaders 與 '--^compact' 標誌搭配使用時，

不會列印變更清單。(不會套用

至新變更清單格式。)

搜尋類型：

--^added 列印已新增的項目。

--^checkout 列印已簽出的項目。

--^changed 列印已變更的項目。

--^copied 列印已複製的項目。

--^replaced 列印已取代的項目。

--^deleted 列印已刪除的項目。

--^localdeleted 列印已在本地刪除的項目。

--^moved 列印已移動的項目。

--^localmoved 列印已在本地移動的項目。

--^percentofsimilarity=<值> 若要將兩個檔案視為相同項目，兩個檔案之間的

相似性百分比。可用於

已在本地移動的搜尋。其預設

值為 20%。

--^txtsameext 在已移動項目搜尋期間，

相似性內容比對程序

只會將具有相同副檔名的

文字檔案納入考量。根據預設，將處理所有

文字檔案。

--^binanyext 在已移動項目搜尋期間，

相似性內容比對程序

會將所有二進位檔案納入考量。根據預設，

只會處理具有相同

副檔名的二進位檔案。

--^private 列印非受控項目。

--^ignored 列印已忽略的項目。

--^hiddenchanged 列印隱藏的已變更項目。(包括

'--^changed')

--^cloaked 列印已隱匿的項目。

--^controlledchanged 此標誌會替代下列選項：

'--^added'、'--^checkout'、'--^copied'、

'--^replaced'、'--^deleted'、'--^moved'。

--^all 此標誌會取代下列參數：

'--^controlledchanged'、'--^changed'、

'--^localdeleted'、'--^localmoved'、'--^private'。

== CMD\_HELP\_WORKSPACESTATUS ==

備註：

'^status' 命令會列印工作區上已載入的變更集，並取得

工作區內已變更的元素。

此命令可用來顯示工作區中暫止的變更；

可使用命令參數搜尋和修改的

變更類型。預設會顯示所有變更，無論是受控

或本地變更。

已在本地移動的搜尋可使用相似性百分比參數 '--^percentofsimilarity' (-^p)，

決定兩個元素是否為相同的項目。

預設值為 20%，但您可調整此值。

可以顯示依用戶端變更清單分組的工作區變更。

'^default' 變更清單包含

其他變更清單中未包含的變更。也就是說，預設變更清單顯示的內容

將取決於指定的搜尋類型標誌。

若要顯示依變更清單分組的變更，則也需要顯示

受控變更 (狀態等於 '^added'、'^checkout'、'^copied'、

'^replaced'、'^deleted' 或 '^moved' 的項目)。因此，顯示變更清單時，

將自動啟用 '--^controlledchanged' 選項。

XML 輸出的預設編碼為 utf-8。

根據預設，除非指定 '--^machinereadable' 或 '--^short' 標誌，

否則此命令將列印目前目錄相對路徑。如果

已指定其中任何一種標誌，此命令將列印絕對路徑。

如果已指定 '--^xml' 標誌，將列印工作區根目錄

相對路徑 (除非也已指定 '--^fp' 標誌，改為列印

絕對路徑)。

範例：

cm ^status

(列印工作變更集以及工作區中所有已變更的項目類型

，但已忽略的項目除外。)

cm ^status --^controlledchanged

(列印工作變更集以及已簽出、已新增、

已複製、已取代、已刪除和已移動的項目。)

cm ^status --^added

(只列印工作變更集以及工作區內已新增的項目。)

cm ^status c:\workspaceLocation\code\client --^added

(以遞迴方式列印工作變更集以及指定路徑下已新增的

項目。)

cm ^status --^changelists

cm ^status --^changelist

(顯示依用戶端變更清單分組的所有工作區變更。)

cm ^status --^changelist=暫止\_待\_檢閱

(顯示名為 '暫止\_待\_檢閱' 的變更清單上的變更)

cm ^status --^changelist=預設 --^private

(顯示 '預設' 變更清單中的變更，顯示私人項目，以及

含有受控變更的項目 (如果有的話)。)

cm ^status --^short --^changelist=暫止\_待\_檢閱 | cm ^checkin -

(簽入 '暫止\_待\_檢閱' 變更清單中的變更。)

cm ^status C:\workspaceLocation --^xml=output.xml

(取得 XML 格式的狀態資訊並在

output.xml 檔案中使用 utf-8。)

cm ^status --^ignored

(顯示所有已忽略的項目。)

輸出：

/main@myrepo@local (^cs:2 - ^head)

^Added

狀態 大小 上次修改 路徑

^Ignored 0 位元組 19 秒鐘之前 out\app.exe

^Ignored 48 秒鐘之前 src

^Ignored 0 位元組 48 秒鐘之前 src\version.c

cm ^status --^ignored --^cutignored

(顯示未忽略上層目錄的已忽略檔案，以及已忽略的

目錄但不顯示其內容。)

輸出：

/main@myrepo@local (^cs:2 - ^head)

^Added

狀態 大小 上次修改 路徑

^Ignored 0 位元組 19 秒鐘之前 out\app.exe

^Ignored 48 秒鐘之前 src

== CMD\_DESCRIPTION\_XLINK ==

建立、編輯或顯示 Xlink 的詳細資料。

== CMD\_USAGE\_XLINK ==

用法：

cm ^xlink [-^w] [-^rs] <xlink\_路徑> / (<變更集規格> | <標籤規格> | <分支規格)>

[<擴充\_規則>[ ...]]

(建立 Xlink。)

cm ^xlink [-^rs] <xlink\_路徑> /<相對\_路徑> (<變更集規格> | <標籤規格> | <分支規格>)

[<擴充\_規則>[ ...]]

(建立指向 /<相對\_路徑> 而非

預設根目錄 / 的唯讀局部 Xlink。)

cm ^xlink -^e <xlink\_路徑> (<變更集規格> | <標籤規格> | <分支規格>)

(編輯 Xlink 以變更目標規格。)

cm ^xlink -^s|--^show <xlink\_路徑>

(顯示 Xlink 資訊，包括擴充規則。)

cm ^xlink -^ar|--^addrules <xlink\_路徑> <擴充\_規則>[ ...]

(將指定的擴充規則新增至 Xlink)。

cm ^xlink -^dr|--^deleterules <xlink\_路徑> <擴充\_規則>[ ...]

(從 Xlink 中移除指定的擴充規則)。

Xlink\_路徑 這是目前工作區中

將裝載 (建立

Xlink 時) 或已裝載 (編輯 Xlink 時) 連結儲存庫的所在目錄。

變更集規格 遠端儲存庫中的完整目標變更集

規格。

這可判斷會在連結儲存庫的工作區中載入

哪個版本和分支。

(使用 'cm ^help ^objectspec' 以深入瞭解變更集

規格。)

標籤規格 遠端儲存庫中的完整標籤規格。

(使用 'cm ^help ^objectspec' 以深入瞭解標籤

規格。)

分支規格 遠端儲存庫中的完整分支規格。

這會使用指定分支指向的

目前變更集。(使用 'cm ^help ^objectspec'

深入瞭解分支規格。)

-^e 編輯現有 Xlink 以變更目標變更集

規格。

-^s | --^show 顯示關於所選 Xlink 的資訊。

-^ar | --^addrules 將一或多個擴充規則新增至所選 Xlink。

-^dr | --^deleterules 從所選 Xlink 刪除一或多個擴充

規則。

擴充\_規則 指定一或多個擴充規則。每個擴充

規則都是一組分支目標分支：

^br:/main/fix-^br:/main/develop/fix

選項：

-^w 代表 Xlink 為可寫入的狀態。這表示

可透過分支自動擴充修改

目標儲存庫內容。

-^rs 相對伺服器。此選項可允許建立

獨立於儲存庫伺服器之外的相對 Xlink。如此

一來，將會自動識別不同伺服器的複寫

儲存庫中建立的 Xlink。

== CMD\_HELP\_XLINK ==

備註：

此命令會對指定的變更集建立 Xlink。依預設會建立唯讀

Xlink。這表示無法修改 Xlink 的工作區中

載入的內容。若要在

Xlink 內容中進行變更，則須改為建立可寫入的 Xlink (使用 '-^w' 選項)。

編輯 Xlink 的目標變更集時，可以使用簡化的

命令語法。如此一來，唯一需要的參數就是

新目標變更集。將不會修改其餘的 Xlink

參數。

分支自動擴充：

在任何可寫入 Xlink 儲存庫中進行變更時 ('-^w' 選項)，必須在目標

儲存庫中建立新分支。新分支

的名稱是根據最上層儲存庫中定義的

簽出分支。若要決定要使用的分支名稱，以下是適用的規則：

1) 執行檢查，確認目標儲存庫中是否存在

具有相同完整名稱的分支：

- 如果存在，這會用來做為簽出分支。

- 如果不存在，會按照下列方式建立分支名稱：

- 目標 Xlink 變更集的分支名稱 + 簽出分支的

簡短名稱 (最後的部分)。

- 如果此分支存在，則會用來做為簽出分支。

- 否則，則會建立分支，且分支基底會設定為

Xlink 變更集。

2) 會在指向 Xlink 儲存庫之新變更集的上層儲存庫的

分支上建立新版 Xlink。

最後，會保留最新版本的完整 Xlink 結構，並包含適當

版本中的最新變更。

範例：

cm ^xlink code\firstrepo / 1@first@localhost:8084

(在目前工作區的 'firstrepo' 資料夾中建立 Xlink，其中

將會裝載儲存庫 'first' 中的變更集 '1'。)

cm ^xlink opengl\include /includes/opengl 1627@includes@localhost:8087

(在目前工作區的目錄 'opengl\include' 中建立唯讀

局部 Xlink，其中儲存庫 'includes' 的變更集 '1627' 中的路徑 '/includes/opengl'

將裝載為根目錄。這表示無論

'/includes/opengl' 中的內容是什麼，都會裝載在 'opengl\include' 中，而

儲存庫的其餘項目則將予以忽略。)

cm ^xlink -^w -^rs code\secondrepo / ^lb:LB001@second@localhost:8084

(在目前工作區的資料夾 'secondrepo' 中建立可寫入

且相對的 Xlink，其中將裝載儲存庫 'second' 的

標籤 'LB001'。)

cm ^xlink code\thirdrepo / 3@third@localhost:8087 ^br:/main-^br:/main/scm003

(在目前工作區的 'thirdrepo' 資料夾中建立 Xlink，其中

將會裝載儲存庫 'third' 中的變更集 '3'。)

cm ^xlink -^e code\secondrepo ^br:/main/task1234@second@localhost:8084

(編輯 Xlink 'code\secondrepo'，在儲存庫 'second'

中連結分支 'main/task1234' 以變更目標儲存庫。)

cm ^xlink --^show code\thirdrepo

(顯示 Xlink 'code\thirdrepo' 的資訊，包括其擴充

規則 (如果有的話))。

cm ^xlink -^ar code\secondrepo ^br:/main-^br:/main/develop ^br:/main/fix-^br:/main/develop/fix

(將兩個擴充規則新增至 xlink 'code\secondrepo'。)

cm ^xlink -^dr code\secondrepo ^br:/main/fix-^br:/main/develop/fix

(從 xlink ' code\secondrepo' 中刪除擴充規則)。

== CMD\_USAGE\_AUTOCOMPLETE ==

用法：

cm ^autocomplete ^install

(將 'cm' 命令完成安裝在 Shell 中。)

cm ^autocomplete ^uninstall

(從 Shell 中解除安裝 'cm' 命令完成。)

cm ^autocomplete --^line <shell\_行> --^position <滑鼠\_位置>

(讓 'shell\_行' 的自動完成建議回到插入於

'滑鼠\_位置'。此命令並非專供最終使用者使用，

但仍記載於文件中，以防您想要為您選擇的 Shell 擴充

自動完成支援。)

shell\_行 使用者已在要求自動完成時

寫入 Shell 的文字行。

在 Bash 中，這位於 COMP\_LINE 環境變數中。

在 PowerShell 中，這位於 $wordToComplete 變數中。

滑鼠\_位置 要求自動完成時的

滑鼠位置。

在 Bash 中，這位於 COMP\_POINT 環境變數中。

在 PowerShell 中，這位於 $cursorPosition 變數中。