

# CA Harvest Software Change Manager

## Command Line Reference Guide

Release 12.5



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

## CA Technologies Product References

This documentation set references the following CA Technologies products:

- CA Harvest Software Change Manager (CA Harvest SCM)
- CA Software Delivery
- CA IT Client Automation (formerly, CA Desktop and Server Management, CA IT Client Manager)
- CA Clarity Agile
- CA Clarity Requirements

## Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- [hudp Command-User Defined Process](#) (see page 161)—Updated this section to include the security hole in the UDP parameter feature.
- [hmvproj Command-Move or Copy Projects](#) (see page 87)—Added this section to include the move or copy projects between databases feature.

## Contact CA Technologies

### Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

### Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to [techpubs@ca.com](mailto:techpubs@ca.com).

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

# Contents

---

## Chapter 1: Introduction 9

Common Options .....	9
User Credentials .....	12
Value Specifications .....	13
Specify Syntax Options .....	14

## Chapter 2: Using the Command Line Utilities 15

caident Command-Get OpenMake Footprint .....	15
haccess Command-Access Report .....	16
hap Command-Approve Package .....	18
har Command-Archive .....	20
hauthsync Command-Authentication Synchronization .....	22
hauthstst Command-Authentication Server API Test .....	23
Basic Inputs Required .....	23
Basic Tests Performed .....	24
Command Format .....	24
External Authentication Connection Options for OpenLDAP .....	25
How to Set Options for LDAP Attribute Names .....	31
How to Use SearchBase DN and Filtering Specifications .....	32
LDAP Parameters for External Usergroup Support .....	33
How to Use OpenLDAP Configuration Files Instead of TLS Startup Parameters .....	34
hcbl Command-Configure Baseline .....	34
hccmrg Command-Concurrent Merge .....	36
hchgtype Command-Change Item Type .....	37
hchu Command-Change Password .....	39
hci Command-Check-in .....	41
hcmpview Command-Compare View .....	46
hco Command-Check-out .....	47
hcp Command-Create Package .....	54
hcpj Command-Copy Project .....	55
hcropmrg Command-Cross Project Merge .....	57
hcrllte Command-Correlate Package .....	60
hcrtpath Command-Create Empty Path .....	61
hdbgctrl Command-Debug Control .....	63
hdelss Command-Delete Snapshot View .....	65
hdlp Command-Delete Packages .....	66

---

hdp Command-Demote Package.....	67
hdv Command-Delete Version .....	68
hexecp Command-Execute Program .....	70
hexpenv Command-Export.....	72
hfatt Command-Form Attachment .....	73
hformsync Command-Update Form Template .....	75
hft Command-Footprint .....	77
hgetusg Command-Get Usage.....	78
himpenv Command-Import.....	79
hlr Command-Load Repository .....	80
hlv Command-List Version .....	82
hmvitm Command-Move Item.....	84
hmvpkg Command-Move Package.....	86
hmvproj Command-Move or Copy Projects.....	87
hmvpth Command-Move Path.....	90
hpg Command-Package Group Manager .....	92
hpkgunlk Command-Unlock Package .....	94
hpp Command-Promote Package .....	95
hpolget Command-Get Password Policy.....	97
hpolset Command-Set Password Policy .....	99
Password Policy.....	100
Policy Configuration File .....	100
hrefresh Command-Refresh Directories .....	107
hredit Command-Edit Repository.....	109
hrepmgr Command-Manage Repository .....	112
Create Repository .....	113
Delete Repository.....	114
Duplicate Repository .....	115
Load Repository .....	116
Rename Repository .....	117
Update Repository Property .....	118
Move Item or Path .....	121
Delete Item or Path.....	122
Rename Item or Path .....	122
Update Item or Path Property.....	123
Create Empty Path .....	124
hri Command-Remove Item .....	125
hrmvpth Command-Remove Path .....	127
hrnitm Command-Rename Item .....	129
hrnpth Command-Rename Path .....	130
hrt Command-Restore.....	132
hsigget Command-Signature File Query.....	134

---

hsigset Command-Signature File Update .....	139
hsmtp Command-SMTP Mail Utility .....	141
Send an Email.....	142
hspp Command-Switch Package .....	142
hsqI Command-RDB Query.....	145
hsv Command-Select Version.....	147
hsync Command-Synchronize Directory from View Path .....	151
htakess Command-Take Snapshot .....	156
hucache Command-Manage Remote Site Cache .....	159
hudp Command-User-Defined Process .....	161
hup Command-Update Package.....	162
husrmgr Command-User Manager .....	165
husrmgr and Password Authentication.....	168
husrunk Command-Unlock User .....	170
svrenc Command-Encrypt User and Password Credentials to a File.....	171

## Appendix A: Return Codes 173

Return Codes and Log Files .....	173
Common Cases .....	174
Special Case - hchu .....	174
Special Case - hco .....	174
Special Case - hexecp .....	175
Return Code Examples .....	175
Windows .....	176
UNIX Bourne Shell .....	176
Perl Script .....	177

## Index 179





# Chapter 1: Introduction

---

**Note:** The instances of Linux in this section refer to both the Linux and zLinux operating environments.

This section contains the following topics:

[Common Options](#) (see page 9)

[User Credentials](#) (see page 12)

[Value Specifications](#) (see page 13)

[Specify Syntax Options](#) (see page 14)

## Common Options

CA Harvest SCM provides you with a command line interface. On UNIX and Linux, the command line utilities serve as the CA Harvest SCM clients.

CA Harvest SCM command line utilities use many common features including options and syntax. These features are described in this chapter, instead of being repeated for every command line utility.

The following options are available for all commands and are not required:

### **-arg**

(Optional) Defines package names that begin with a hyphen (-).

This option has the following format:

```
command ... -arg="-packagename"
```

### **Example:**

```
hcp ... -arg="-mypackagename"
```

### **-h**

(Optional) Displays a brief description of help on the command usage.

**-i *inputfile.txt***

(Optional) Defines the file name of an input file that contains all command line parameters. It is typically used to prevent login credentials from being displayed on the console. This file is read at program execution. The following considerations apply to the input file:

- The file must be a regular text file.
- The file cannot be encoded or encrypted.
- The input file can have any name and extension, and can reside in any directory.
- The file must be available before the command is executed.
- The options and arguments you specify in the file are the same ones available for the command line prompt. They are specified in the same one-line format as the other CA Harvest SCM command line programs.
- You cannot specify some options on the command line and others in an input file; they must all be included in one or the other.
- The input file maximum size is limited only by what the operating system can handle.

**-di *inputfile.txt***

(Optional) Defines the file name of an input file that contains all command line parameters. It is typically used to prevent login credentials from being displayed on the console. This file is read at program execution and then deleted as soon as the file is parsed. If you use this option, you cannot also use the -i option. The following considerations apply to the input file:

- The file must be a regular text file.
- The file cannot be encoded or encrypted.
- The file can have any name and extension, and can reside in any directory.
- The file must be available before the command is executed.
- The options and arguments you specify in the file are the same ones available for the command line prompt. They are specified in the same one-line format as the other CA Harvest SCM command line programs.
- You cannot specify some options on the command line and others in an input file; they must all be included in one or the other.
- The file maximum size is limited only by what the operating system can handle.

**-o filename**

(Optional) Defines the file name of a file to receive the output log. A file name is required; if one is not supplied, the file *utilityname.log* is created in the current working directory (*utilityname* is the name of the command being executed). If a file of the same name already exists, the file will be replaced; log information is not appended to the existing file.

**Note:** This option cannot also be used with -oa.

**-oa filename**

(Optional) Defines the file name of a file to receive the output log. A file name is required; if one is not supplied, the file *utilityname.log* is created in the current working directory (*utilityname* is the name of the command being executed). If a file of the same name already exists, log information is appended to the existing file.

**Note:** This option cannot also be used with -o.

**-prompt**

(Optional) Specifies that the user should be prompted for a user name (-usr) and password (-pw) at runtime. This option is typically used to hide login credentials from being exposed in plain text on the command line. Including this option overrides use of -usr and -pw. This option also overrides the -eh option.

**-v**

(Optional) Specifies that verbose option be used to display all relevant variables being used for the command execution.

**UNIX and Linux:** Using verbose mode, you can confirm whether the command should actually be process.

**Windows:** Verbose mode simply writes the appropriate values to the output file.

**Note:** An exception is *hsigget*, which uses the -v option to generate a report with a pre-selected subset of attributes related to version information, including file name, status, tag, version, and package.

**-wts**

(Optional) Specifies that the timestamp of command line execution be recorded in the output log.

## User Credentials

You can determine whether the password is hidden for a command line utility. Most of the commands discussed in this guide allow for the specification of CA Harvest SCM user credentials or user credentials for a remote operating system (for example, for a file agent login).

If the CA Harvest SCM server or file agent process does not allow command line utilities to pass clear-text passwords and if the command line is determined to be passing clear-text passwords, the CA Harvest SCM server or file agent process will reject the create session or file agent login request.

Password information is considered hidden on command line utilities if any of the following are true:

- The command line specifies one of the input file options (for example, `-i` or `-di`).
- The input file option is *not* specified and the password option (for example, `-pw`, `-rpw`) is *not* specified on the command line.

### Examples: Determine User Credentials

The following command line statements show examples of how user credentials are determined.

The following statement is *not* considered to have a hidden password because the password option is explicitly specified on the command line:

```
hco -b brokername -usr username -pw password ...
```

The following statement is considered to have a hidden Remote OS user password, because even though clear text credentials are being specified, the password is not explicitly specified on the command line. In this case the command will prompt for the user's password information.

```
hco -b brokername ... -rm remote_computer -rusr username ...
```

The following statement is considered to have a hidden CA Harvest SCM user password because an encrypted credential file is used to specify the user credentials for the command.

```
hco -b brokername -eh encrypted_credential_file ...
```

The following statement is considered to have a hidden CA Harvest SCM user password because the options are specified in the context of an input file:

```
hco -i inputfile
```

*inputfile* contains the following options:

```
-b brokername -usr username -pw password ...
```

The following statement is *not* considered to have a hidden CA Harvest SCM user password because the password option is specified explicitly, even though the user name and password specifications are overwritten as a result of the -prompt option.

```
hco -b brokername -usr username -pw password -prompt ...
```

The following statement is *not* considered to have a hidden Remote OS user password, because even though an encrypted credential file is specified, the remote password option is specified.

```
hco -b brokername ... -rm remote_computer -er encrypted_credential_file -rpw password ...
```

The following statement is *not* considered to have a hidden CA Harvest SCM user password, because even though an encrypted credential file is specified, the password option is specified.

```
hco -b brokername -eh encrypted_credential_file -pw password ...
```

## Value Specifications

If the value for a command contains one or more spaces, enclose the value with quotation marks (" "). For example, if a view path name contains a space, such as Test Repository, and you specify the following, the process fails:

```
... -vp \Test Repository
```

Instead, enclose the value in quotation marks, as follows:

```
... -vp "\Test Repository"
```

On the command line, the shell interprets the backslash (\) as a quotation mark if a quotation mark follows the backslash (\"). This sequence leads to parsing errors; other CA Harvest SCM errors may occur depending on the order of the arguments on the command line. To avoid this confusion, replace a single backslash with double backslashes preceding a quotation mark, or do not use a single backslash preceding a quotation mark.

For example, instead of:

```
-vp "\Test Repository\"
```

use:

```
-vp "\Test Repository\\"
```

or

```
-vp "\Test Repository"
```

Command line parameters such as `-vp "\Test Repository\"` in an input file, when using option `-i` or `-di`, will be parsed correctly.

**Note:** CA Harvest SCM does not interpret the backslash (\) as a special character.

On UNIX or Linux, view paths that contain subsequent paths must be enclosed in quotation marks even though they contain no spaces. For example:

```
-vp "\Testrepository\path1"
```

## Specify Syntax Options

To specify command options, use one of the following syntax models:

```
-optionname optionarg1 [... optionargn ]
```

```
-optionname=optionarg1 [... optionargn ]
```

**Note:** To specify command line argument names that begin with a hyphen (-), use the `-arg` option. For more information about these options, see [Common Options](#).

# Chapter 2: Using the Command Line Utilities

---

**Note:** The instances of Linux in this section refer to both the Linux and zLinux operating environments.

## caident Command-Get OpenMake Footprint

caident.exe browses recursively through a directory and captures the OpenMake footprint information for all the libraries and executables that are built using the OpenMake build management tool in an XML readable format. For any other executable or dll not built using OpenMake, the output is empty. Every time caident is executed in the same directory, the XML file is created. Information is not appended to the existing file in the directory.

This command has the following format:

```
caident [-dir path] [-r]] [-f filename] [-xml filename] [-o filename] [-v] [-h]
```

**-dir *path***

(Optional) Specifies the full path of the directory from which the executables are identified. If the option is not specified, the default value is the current working directory.

**-r**

(Optional) Finds the executables recursively through the path name given in -dir option. By default, caident searches for the executables in the directory specified by the -dir option on the current working directory if the -dir option is not specified.

**-f *name***

(Optional) Specifies the filename or pattern that provides the identification information. By default caident assumes wildcards (\*.\*) for the list of filenames.

**-xml *name***

(Optional) Specifies the output filename for the executable identifier XML. By default, the file name is caident.xml.

**-o *outputfilename***

(Optional) Specifies the output log filename. By default, the filename is caident.log and is created in the directory from which the caident.exe is invoked.

**-v**

Specifies verbose output.

**-h**

Specifies the instructions to use caident.

## haccess Command-Access Report

The haccess command is a toolkit for the CA Harvest SCM administrator who needs to print out detailed access information for projects in CA Harvest SCM. You can specify the wildcard asterisk (\*) character to display information on multiple projects. You can generate five types of reports using haccess, but only one report can be displayed at a time.

To generate a report, you must have the necessary access permissions as listed in the following table:

Report	Access Required
Project	SuperUser, AdminEnvironment, ViewEnvironment
Repository	SuperUser, AdminRepository, ViewRepository
Harvest	SuperUser, SecureHarvest
Usergroup	SuperUser, AdminUser, ViewUser
FormType	SuperUser, AdminFormType, ViewFormType

This command has the following format:

```
haccess {-b name} {-usr username -pw password} [-en name] [-rn name] [-ha]
[-ug name] [-ft name] [-i inputfile.txt | -di inputfile.txt] [-eh filename] [-prompt]
[-o filename | -oa filename] [-arg] [v] [-wts] [-h]
```

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-usr username**

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw password**

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.



**-en *name***

(Optional) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks. The limit of projects that can be displayed is 5000.

**-rn *name***

(Optional) Specifies the name of the repositories to report on. The limit of repositories that can be displayed is 5000.

**-ha**

(Optional) Specifies the highest level of access.

**-ug *name***

(Optional) Specifies the name of the user groups to report on.

**-ft *name***

(Optional) Specifies the names of the form types to report on.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**Examples: haccess command**

The following example reports the access setup on a given project and its sub-objects:

- List the group names that have Secure, Update, View, and Use Access to the project.
- List the group names that have Update and Update Package access to each state in the project.
- List the group names that have Execute Access to each process in each state in the project.

```
haccess -b BrokerName -usr UserName -pw Password -en "Project*"
```

The following example reports the access setup on a given repository and its sub-objects such as items and item paths. Repositories are identified by name.

- List the group names that have Secure, Update, and View Access to the repository.

```
haccess -b BrokerName -usr UserName -pw Password -rn "Repository*"
```

The following example reports the access setup at the highest level:

- List the usernames that are in Administrator user group.
- List the group names that have Global Secure, Admin Project, Admin Repository, Admin User, Admin Form Type, View Project, View Repository, View User, View Form Type accesses.

```
haccess -b BrokerName -usr UserName -pw Password -ha
```

The following example reports all access that a given user group has been granted:

- List the highest-level methods that group has been granted: Global Secure, Admin Project, Admin Repository, Admin User, Admin Form Type, View Project, View Repository, View User, and View Form Type.
- List the projects that group has Secure or Update, View, or Execute access.
- List the processes that group has execute access with its state.
- List the state that group has Update and Update Package access in any state.
- List the repositories that group has Secure, Update, View, or Execute access.
- List the items/item paths that group has No View access.

```
haccess -b BrokerName -usr UserName -pw Password -ug "UsrGrp Name"
```

The following is an example of the access setup on a given form type:

```
haccess -b BrokerName -usr UserName -pw Password -ft "Form Type"
```

## hap Command-Approve Package

The hap command executes the approval process specified on the command line, or the first approve process in the current context if no process name is specified. The process definition determines the default behavior, process access, and execution of linked processes. Other aspects of the process can be specified through command parameters.

This command has the following format:

```
hap package_name1 package_name2... {-b name -en name -st name} [-v] [-pn name] [-c text] {-usr username -pw password} [-rej] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]package_name ...
```

Specifies the names of one or more packages to approve.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *name***

(Required) Specifies the name of the state in which the user is operating.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-c *text***

(Optional) Defines a comment about the approval.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-rej**

(Optional) Specifies that the package is to be rejected. If this option is not specified, the package is approved.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## har Command-Archive

The CA Harvest SCM archive utility lets administrators move data from the CA Harvest SCM database tables to another location for archive purposes.

The following rules apply to the archive function:

- A CA Harvest SCM broker and server must be running on the CA Harvest SCM installation on which you want to run the archive.
- The archive function stores files at the project level. You can specify one or more projects to archive in one execution.
- Only inactive projects can be archived.
- Only CA Harvest SCM administrators can run the archive utility.
- Life cycle templates cannot be archived.
- If you archive to the same destination and file name as an existing archived project, the existing archive is overwritten; the new archive will not be appended to the existing one.

After a successful archive execution, two files are created in the archive destination location:

- The .arv file is a text file that stores the logical records: harVersions, harBranch, harItemRelationship, harVersionInView, harVersionData.
- The .dat file is a binary file that stores the physical data.

**Important!** Do not delete or modify the .arv or .dat files if you plan to restore the project. Deletion or modification of these files will result in restoration failure.

After archiving a project, the project status is automatically set to Archive.

**Note:** Administrators can use the restore (hrt) utility to move data back to the original database.

This command has the following format:

```
har {-b name} {-f} [-m computername -rport portnumber] [-musr username -mpw password] {-usr username -pw password} [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename] [-er filename] [-o filename | -oa filename] [-arg] [-v] [-wts] [-h] project_name
```

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-f**

(Required) Defines the archive file.

**-m *computername***

(Optional) Specifies the name of the destination computer for the archive, if you are archiving to a remote computer.

**-rport *portnumber***

(Optional) Specifies the port number of the destination computer for the archive, if you are archiving to a remote computer.

**-musr**

(Optional) Specifies the user name of the destination computer for the archive, if you are archiving to a remote computer.

**-mpw**

(Optional) Specifies the password of the destination computer for the archive, if you are archiving to a remote computer.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-er *filename***

(Optional) Specifies the encrypted remote username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility. -er overrides -musr and -mpw options.

***project\_name***

Specifies the name of the project to archive.

**More information:**

[hrt Command-Restore](#) (see page 132)

## hauthsync Command-Authentication Synchronization

This command sends an authentication synchronization request to the broker. For CA Harvest SCM users that also exist on the external authentication server, synchronization refreshes the values of Real Name, Phone#, Phone Extension, Fax#, and Email. For CA Harvest SCM users who do not exist on the external authentication server, the users are ignored and their properties not updated.

**Note:** To execute this command, the user must be member of the CA Harvest SCM Administrators group or have Admin User Access rights.

Upon completion, the command issues a summary message indicating the number of CA Harvest SCM users ignored and updated. For example:

External authentication synchronization summary: Users ignored: 2; Users updated: 5.

This command has the following format:

```
hauthsync {-b name} {-usr username} {-pw password} [-v] [-prompt] [-i inputfile.txt |  
-di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Optional) Defines the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Optional) Defines the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hauthtst Command-Authentication Server API Test

This command line utility tests the Authentication Server API independently of a CA Harvest SCM broker or RTserver.

### Basic Inputs Required

The basic inputs required to run the test utility include the following inputs:

- The authentication server type (for example, openldap)
- Connection information for the authentication server. For the LDAP external authentication option, this configuration would consist of the following:
  - LDAP server name
  - LDAP server port
  - LDAP security mode (for example, SSL, TLS, or none [unencrypted])
  - LDAP bind distinguished name (DN) to use for querying the LDAP directory
  - Password credentials file used to authenticate the bind DN on the directory
  - Optional search DN
  - Optional filter DN
  - Timeout factor user lookup on the SearchBase DN
  - Certificate Authority (CA) certificate and client certificate key files as required by the directory server
  - Directory attribute name adjustments for particular directory servers
- Number of times to repeat the test
- List of users and their credentials to authenticate. The hauthtst utility retrieves the following attributes for the specified users:
  - User information attributes (for example, Full Name, Email address)
  - User impersonation attributes for UNIX systems

## Basic Tests Performed

The hauthtst utility performs the following basic tests of the CA Harvest SCM Authentication Server API, in the order shown:

1. Connects to the authentication server using the administrative credentials.  
This step creates the authentication server “session” and authenticates the administrative user against the authentication server. For example, in LDAP terms, this operation initializes the LDAP API and authenticates the LDAP directory server with the bind DN and password.
2. Retrieves attributes associated with the authentication server (for example, server type and directory type).
3. Retrieves global password policy attributes.
4. Performs the following operations on the specified list of users and their credentials:
  - Authenticates each user and credentials against the authentication server.
  - Retrieves basic property attributes for the user.
  - Retrieves impersonation attributes for the user (for example, for UNIX).
5. Disconnects the authentication server session.  
For example, in LDAP terms, this unbinds the authentication server session from the user requesting directory data.
6. If an iteration count is specified, repeats the previous steps the number of times specified.

## Command Format

This command has the following format:

```
hauthtst [-authmode=internal | openldap] [list_of_user_accounts] [-r iterations]  
external_authentication_connection_options [-prompt] [-i inputfile.txt]  
| -di inputfile.txt] [-o filename | -oa filename] [-wts] [-h]
```

### **-authmode={internal|openldap}**

(Optional) Specifies the authentication mode that the remote CA Harvest SCM agent uses to authenticate users' logon credentials.

#### **openldap**

Specifies an LDAP v3-compliant directory server (LDAP server).

#### **internal**

Specifies internal (CA Harvest SCM) authentication.

**Default:** internal



### ***list\_of\_user\_accounts***

(Optional) Specifies one or more user accounts used to test login to the authentication server. The format for each user specifications is:

`"username;password"`

or

`"username"`

Separate user account entries with spaces.

If you specify only the username, use the `-prompt` option to prompt for and retrieve the password for the user account.

### ***-r iterations***

(Optional) Specifies the number of times to repeat the test.

### ***external\_authentication\_connection\_options***

Defines the connection options for the authentication mode you are using. For details, see External Authentication Connection Options for OpenLDAP.

## External Authentication Connection Options for OpenLDAP

**Note:** Always enclose a value in quotation marks (" ") when it contains spaces.

**`-ldapservers="hostname1[:port1] [hostname2[:port2] [hostname3[:port3]...]"`**

Defines one or more host names of the LDAP server. For example:

`-ldapservers=hostname1`

You can optionally define the port number to use on each host, by entering the host name in the form `hostname:port`. For example:

`-ldapservers=hostname2:389`

**Important!** If used, the `:port` option overrides the port number provided in the `-ldapport` parameter.

You can specify a list of host names, separated by spaces and enclosed in quotation marks. Each host can optionally follow the form `hostname:port`. For example:

`-ldapservers="hostname1:389 hostname2 hostname3:389"`

When the `-ldapservers=hostname[:port]` parameter defines multiple host names, the CA Harvest SCM server or agent connects to the first available LDAP server in the list.

**Limits:** 255 characters

**-ldapport=*portnumber***

(Optional) Specifies the port number for the LDAP server computer. This parameter is used if the LDAP port number is not specified in the `-ldapserver=hostname[:port]` parameter.

**Default:** If `-ldapmode=ssl`, then the default is 636; otherwise, the default is 389.

**Minimum:** 1

**Maximum:** 9999

**-ldapbinddn=*distinguished\_name***

Defines the LDAP initial bind DN to the LDAP server. For all authentication operations, only the initial DN is used to bind to the LDAP directory.

**Default:** None

**Limits:** 255 characters

A sample entry follows:

`-ldapbinddn="cn=john22,ou=users,ou=north america,dc=abccorp,dc=com"`

**-ldapbindpw=*password***

Defines the password for the LDAP distinguished name used for the initial bind. Do *not* enter spaces.

If you do not specify a password, an empty password is used.

**Important!** The value you specify is saved in clear text. For security reasons, we recommended that you provide the password using the `-ldapbindpwfile={filename}` parameter.

**Limits:** 255 alphanumeric characters

**-ldapbindpwfile=*filename***

Specifies either the file name or the complete path name for the file containing the encrypted password.

**Default:** If you do not specify a complete path name, the default path name is used. The default path name is the CA Harvest SCM installation directory: `%CA_SCM_HOME%` on Windows or `$CA_SCM_HOME` on UNIX or Linux.

If the `-ldapbindpwfile=filename` parameter is specified, it overrides the value provided in `-ldapbindpw=password`.

**Note:** Use the command line utility `svrenc` to generate a file containing encrypted username and password values. The password value is the only one used; the username value is ignored. For details, see the description of the `svrenc` command in this chapter.

### **-ldapfilter=*search\_filter***

(Optional) Defines an RFC-2254-compliant search filter for locating a user. When a user attempts to log in to CA Harvest SCM, this filter is used to search for the user on the LDAP server.

**Default:** (&(objectclass=person)(uid=<uid>))

**Note:** The complete expression for the search filter that your LDAP server uses may differ from the default value, depending on how your LDAP server is configured. For details, see your system administrator.

### **(uid=<uid>)**

Specifies the LDAP user attribute name (*uid*) and placeholder (<uid>) used in the search.

#### **uid**

Defines your LDAP server's attribute name for user name. This value must match the value specified for your LDAP server by the LDAP user attribute name parameter, -ldapattrusname=*attribute\_name*.

### **-ldapattrusname=*attribute\_name***

(Optional) Defines your LDAP server's user attribute name for user name, for example:

-ldapattrusname=cn

-ldapattrusname=uname

### **Examples: Defining the LDAP user attribute name**

Using the default search filter, if -ldapattrusname=cn for your LDAP server, then the search filter is:

(&(objectclass=person)(cn=<cn>))

If -ldapattrusname=uname for your LDAP server, then the search filter is:

(&(objectclass=person)(uname=<uname>))

#### **<uid>**

Identifies a literal constant used as a placeholder. Enter your LDAP server's attribute name for user name with the angle brackets (< >), literally as shown.

For example, when a user attempts to log in to CA Harvest SCM, <uid> is replaced dynamically with the user's user name, and the directory is searched for this user.

### Examples: How the Search Filter is Used

The following examples use the default search filter and use the setting `-ldapattrusrname=uid`.

When the user amy33 attempts to log on, the search filter used to locate this user is:

```
(&(objectclass=person)(uid=<amy33>))
```

When the user john22 attempts to log on, the search filter used to locate this user is:

```
(&(objectclass=person)(uid=<john22>))
```

### **-ldapsearchtimeout=seconds**

(Optional) Defines the number of seconds to search for a user in the LDAP directory; for example, when a user attempts to log in to CA Harvest SCM.

**Default:** 60 (1 minute)

**Limits:** 20 digits

### **-ldapbasedn=base\_distinguished\_name**

Defines the base distinguished name (DN) used when searching on the LDAP server. For example:

```
-ldapbasedn="ou=users,ou=north america,dc=abccorp,dc=com"
```

**Default:** None

**Limits:** 255 characters

### **-ldapmode={none|tls|ssl}**

Specifies the security mechanism to use for authenticating CA Harvest SCM users:

#### **none**

Specifies no security mechanism.

**Important!** If you specify none (no encryption), then user credentials and all other information exchanged between CA Harvest SCM and the LDAP server are transmitted in clear-text mode.

#### **tls**

Specifies Transport Layer Security.

**Note:** Specify TLS *only* if your LDAP server supports StartTLS.

#### **ssl**

Specifies Secure Socket Layer.

**Default:** tls

**-ldapdebuglevel=*debuglevel***

(Optional) Specifies the LDAP debugging level when running this utility. When you specify a valid non-zero value, additional debugging information is printed to the command's output file. This option is typically used when investigating LDAP connectivity problems.

**Default:** 0 (no debugging)

Possible values include the following:

```
LDAP_DEBUG_TRACE 0x0001
LDAP_DEBUG_PACKETS 0x0002
LDAP_DEBUG_ARGS 0x0004
LDAP_DEBUG_CONNS 0x0008
LDAP_DEBUG_BER 0x0010
LDAP_DEBUG_FILTER 0x0020
LDAP_DEBUG_CONFIG 0x0040
LDAP_DEBUG_ACL 0x0080
LDAP_DEBUG_STATS 0x0100
LDAP_DEBUG_STATS2 0x0200
LDAP_DEBUG_SHELL 0x0400
LDAP_DEBUG_PARSE 0x0800
LDAP_DEBUG_CACHE 0x1000
LDAP_DEBUG_INDEX 0x2000
LDAP_DEBUG_NONE 0x8000
LDAP_DEBUG_ANY -1
```

**-tlstrcertfile=*filename***

(Optional) Defines the complete path name of the TLS trusted certificate file. This parameter specifies the PEM-format file containing certificates for the Certificate Authorities (CAs) that the LDAP client (CA Harvest SCM remote agent or server) will trust. The certificate for the CA that signed the LDAP server certificate must be included in these certificates. If the signing CA was not a top-level (root) CA, certificates for the entire sequence of CAs from the signing CA to the top-level CA should be present. Multiple certificates are simply appended to the file; the order is not significant.

You can also define the TLS trusted certificate file in the OpenLDAP configuration file (for example, on UNIX, in the \$HOME/.ldaprc file) using the following parameter: TLS\_CACERT *filename*

**-tls-certfile=*filename***

(Optional) Defines the complete path name of the TLS client certificate file.

You can also define this certificate file in the OpenLDAP configuration file (for example, on UNIX, in the \$HOME/.ldaprc file) using the following parameter:  
TLS\_CERT *filename*

**-tls-keyfile=*filename***

(Optional) Defines the complete path name of the TLS private key associated with the client certificate file.

You can also define this key in the OpenLDAP configuration file (for example: on UNIX, in the \$HOME/.ldaprc file) using the following parameter: TLS\_KEY *filename*

**Important!** Private keys are sensitive data and are usually password-encrypted for protection. However, the current LDAP API implementation does not support encrypted keys. Therefore, you cannot encrypt the key, and you must protect the file containing the key.

**-ldapattrsrunixuid=*attribute\_name***

(Optional) Defines your LDAP server's user attribute name for UNIX user ID.

**Default:** uidNumber

**-ldapattrsrunixgid=*attribute\_name***

(Optional) Defines your LDAP server's user attribute name for UNIX primary group ID.

**Default:** gidNumber

**-ldapattrsrunixhomedir=*attribute\_name***

(Optional) Defines your LDAP server's user attribute name for UNIX home directory.

**Default:** homeDirectory

**-ldapattrsrunixshell=*attribute\_name***

(Optional) Defines your LDAP server's user attribute name for UNIX shell.

**Default:** shell

**-ldapattrsrfullname=*attribute\_name***

(Optional) Defines your LDAP server's user attribute name for Full Name.

**Default:** fullName

**-ldapattrusrphone=*attribute\_name***

(Optional) Defines your LDAP server's user attribute name for Phone Number.

**Default:** telephoneNumber

**-ldapattrusrphoneext=*attribute\_name***

(Optional) Defines your LDAP server's user attribute name for Phone Extension.

**Default:** telephoneExtension

**-ldapattrusrfax=*attribute\_name***

(Optional) Defines your LDAP server's user attribute name for Fax Number.

**Default:** facsimileTelephoneNumber

**-ldapattrusremail=*attribute\_name***

(Optional) Defines your LDAP server's user attribute name for eMail.

## How to Set Options for LDAP Attribute Names

Verify that your CA Harvest SCM LDAP attribute names are the same as the attribute names used by your LDAP server. CA Harvest SCM provides configuration parameters for the following LDAP attribute names:

-ldapattrusrname

-ldapattrusrfullname

-ldapattrusrphone

-ldapattrusrphoneext

-ldapattrusrfax

-ldapattrusremail

-ldapattrusrunixuid

-ldapattrusrunixgid

-ldapattrusrunixhomedir

-ldapattrusrunixshell

**Example: Changing the user attribute name for user name**

To change the user attribute name for user name to sAMAccountName, use the following parameter:

```
-ldapattrusrname=sAMAccountName
```

#### Example: Configuring CA Harvest SCM to not retrieve an attribute value

You can configure CA Harvest SCM so that it does not retrieve the value for a specific attribute by using an empty string as the attribute name. For example, to prevent CA Harvest SCM from retrieving users' phone numbers, specify the following:

```
-ldapattrusrphone=""
```

## How to Use SearchBase DN and Filtering Specifications

The LDAP Authentication Server API uses an OpenLDAP filter specification to help select the proper user account container from a set of all entries associated with the SearchBase DN (for example, the `-ldapbasedn` parameter). The filter specification uses a *pre-fix* operator format, rather than the more common *in-fix* operator format. For example, the default filter is the following expression:

```
(&(objectclass=person)(uid=<uid>))
```

This expression means: Search for directory entries that meet both of the following criteria:

- The user name attribute (uid) value equals the requested user identifier
- The object class attribute value equals “person”
- In some cases, this level of filtering is insufficient. For example, on certain directories that have *computer* containers and *user account* containers, the previous filter may return a computer container entry for the user rather than the user account container. In such cases, refine the filter to exclude computer node entries, for example:

```
(&(!(objectclass=computer))(&(objectclass=person)(uid=<uid>)))
```

This expression means: Search for directory entries that meet all of the following criteria:

- The user name attribute (uid) value equals the requested user identifier
- The object class attribute value equals “person”
- The object class attribute value does *not* equal “computer”



The time required for the search depends mostly on the selection of the Search Base DN. For example, the following expression searches an entire domain:

```
DC=mydomain,DC=com
```

When searching large domains, you can reduce the search time by limiting the search to a specific portion of the entire domain, for example:

```
OU=North America, DC=mydomain,DC=com
```

## LDAP Parameters for External Usergroup Support

You can use the following parameters to define LDAP support for external user groups:

**Note:** Always enclose a value in quotation marks ( " ") when it contains spaces.

### **-externalgroupenabled=1 or 0**

(Optional) Use the following values to enable or disable your LDAP server's user group for external authentication:

- 1 enables your LDAP server's user group for external authentication.
- 0 disables your LDAP server's user group for external authentication.

### **-ldapgrpfilter=(&(objectclass=<objectclass of usergroup>)(usergroup-attribute-name=<placeholder>))**

(Optional) Defines a group filter for locating a particular user group in the ldap server.

### **usergroup-attribute-name**

Attribute of the usergroup used in -ldapattrusrgrpname=*attribute\_name*.

### **<placeholder>**

Identifies a literal constant placeholder for *usergroup-attribute-name*. Enter exactly the same value as *usergroup-attribute-name* and enclose the value with angle brackets (< >), as shown in the following example.

### **Example: Use -ldapattrusrgrpname=cn as the LDAP Server**

In this example, if the objectclass of usergroup is group, and if -ldapattrusrgrpname=cn is used for your LDAP server, then the group filter is the following:

```
(&(objectclass=Group)(cn=<cn>))
```

**-ldapattrusringrp=*attribute\_name***

(Optional) Defines your LDAP server's attribute that evaluates members/users of a group, for example:

-ldapattrusringrp=*member*

**-ldapattrgrpinsusr=*attribute\_name***

(Optional) Defines your LDAP server's attribute that evaluates groups of a user, for example:

-ldapattrgrpinsusr=*member\_of*

## How to Use OpenLDAP Configuration Files Instead of TLS Startup Parameters

On UNIX, instead of specifying TLS parameters as command line or .arg file parameters, you can optionally define the TLS certificate and key specifications in an OpenLDAP configuration file located in \$HOME/.ldaprc.

For example, you can add the following TLS parameters to the .ldaprc file:

**TLS\_CACERT "*filename*"**

Defines the complete path name of the TLS trusted certificate file. For details about this file, see the description of the -tlstrcertfile=*filename* parameter.

**TLS\_CERT "*filename*"**

Defines the complete path name of the TLS client certificate file. For details about this file, see the description of the -tlscertfile=*filename* parameter.

**TLS\_KEY "*filename*"**

Defines the complete path name of the TLS private key associated with the client certificate file. For details about this file, see the description of the -tlskeyfile=*filename* parameter.

## hcbl Command-Configure Baseline

The hcbl command enables you to configure a project baseline.

This command has the following format:

```
hcbl {-b name -en name -rp name} [-usr username] [-pw password] [-ss name] [-add] [-rmr]
[-rdo] [-rw] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename] [-st
name] [-v] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-rp *name***

(Required) Specifies the repository that will be added or removed from the project's baseline.

**-usr *username***

(Optional) Defines the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Optional) Defines the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-ss *name***

(Optional) Specifies the name of the snapshot view to which the repository belongs. If this option is not specified, the initial repository view is selected.

**-add**

(Optional) Adds the specified repository to the baseline.

**-rmr**

(Optional) Removes the specified repository from the baseline.

**-rdo**

(Optional) Sets the specified repository in the baseline to read-only.

**-rw**

(Optional) Sets the specified repository in the baseline to write access.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-st *name***

(Optional) Specifies the state to use for Version Control Interface (VCI) binding corrections.

## hccmrg Command-Concurrent Merge

The hccmrg command executes the concurrent merge process specified on the command line or the first concurrent merge process in the current context if no process name is specified. The process properties determine the default behavior, process access, and execution of linked processes. Other aspects of the process can be specified through command parameters read from the command line.

**Note:** At least one item must be merged for a post-link process to execute.

This command has the following format:

```
hccmrg {-b name -en name -st name -p name} {-usr username -pw password} [-v] [-mc]
[-ma] [-tt] [-tb] [-pn name] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh
filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Optional) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *name***

(Required) Specifies the name of the state in which the user is operating.

**-p *name***

(Optional) Specifies the CA Harvest SCM package name, read from the command line or a signature file.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-mc**

(Required) Specifies to merge conservatively, which creates a merge-tagged version, regardless of the contents of the versions. An exception is when the branch version is the latest version in the view; in this case, it is closed and a normal version is created.

**-ma**

(Optional) Specifies to merge aggressively, which creates a merge-tagged version only when conflicts are found. If no conflicts are found, the branch and trunk versions are merged to create a normal version.

**-tt**

(Optional) Specifies to take the trunk version, which automatically selects the trunk (target) to create the final version, without comparing the contents of the versions. Using this option closes the branch, but does not create any new versions on the trunk.

**-tb**

(Optional) Specifies to take the branch version, which automatically selects the branch (source) to create the final version, without comparing the contents of the versions. Using this option creates a normal version on the trunk and closes the branch.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hchgtype Command-Change Item Type

The hchgtype command is an administrator only utility that converts the type of an item, specified by file extension, in the CA Harvest SCM repository between text and binary. The user must be a member of the CA Harvest SCM Administrator user group to execute hchgtype.

**Important!** Converting binary data such as applications, DLLs, libraries, or documents that cannot be viewed in a simple text editor, to text could result in a loss of information. Therefore, backing up the database is recommended before you execute item conversion.

This command has the following format:

```
hchgtype [ext1 ext2... | itemname1 itemname2...] {-b name -rp path} [-v]
{-usr username -pw password} {-bin | -txt} [-ext] [-g] [-prompt] [-i inputfile.txt
| -di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**ext... | itemname...**

(Optional) Specifies file extension types or specific items to convert between text and binary. If extensions are specified, you must include the -ext option.

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-rp path**

(Required) Specifies the path to the CA Harvest SCM repository.

**-usr username**

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw password**

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-bin**

(Required: -bin and -txt are mutually exclusive and one is required.) Specifies conversion of the item to binary format.

**-txt**

(Required: -bin and -txt are mutually exclusive and one is required.) Specifies conversion of the item to text format.

**-g**

(Optional) Specifies that the entire repository (specified by the -rp option) will be searched. If this option is *not* specified, then only the specific repository path is searched.

**-ext**

(Optional) Specifies that the arguments to hchgtype are to be treated as extensions: exe, bin, and so on. All files in the area being searched that have a matching extension are converted. If this option is not specified, the arguments to hchgtype are treated as full file names.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**Examples: hchgtype command**

To change the type of the file named file1.txt located in the repository path \Rep1\DirA\DirB to text, the command line is:

```
hchgtype file1.txt -b brokerName -rp "\Rep1\DirA\DirB" -usr username -pw password -txt
```

To change the type of all files with a .txt extension in the repository named Rep1 to binary, the command line is:

```
hchgtype txt -b brokerName -rp "\Rep1\DirA\DirB" -usr username -pw password -bin -ext -g
```

**Note:** Because the -g option is used in this example, specifying \DirA\DirB causes the same behavior as specifying simply "\Rep1" for the -rp argument.

To change the type of all files with a .txt extension located in "\Rep1\DirA" to text, the command line is:

```
hchgtype txt -b brokerName -rp "\Rep1\DirA" -usr username -pw password -txt -ext
```

## hchu Command-Change Password

The hchu command changes a password. This command line may be used by users to change their own passwords or by a CA Harvest SCM Administrator to change the password of other users.

If the -npw option is not specified on the command line, the user is prompted for a new password. Passwords are validated against the CA Harvest SCM Password Policy.

The following considerations apply to authentication:

- When you are using internal authentication (CA Harvest SCM authentication), changing passwords is enabled.

- When you are using external authentication, such as Microsoft Active Directory, this functionality is disabled; user passwords must be changed using methods provided by the authentication server. For example, if using Microsoft Active Directory, you can change passwords by entering Ctrl+Alt+Del and using the Change Password option.
- When you are using external authentication, changing passwords fails with an error message.

This command has the following format:

```
hchu {-b name} [-usr username] [-pw password] [-npw new_password]  
[-ousr other_username] [-v] [-prompt] [-i inputfile.txt | -di inputfile.txt]  
[-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Optional) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Optional) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-npw *new\_password***

(Optional) Defines a user's new password.

**-ousr *other\_username***

(Optional) Specifies the Administrator option to change the password of another user name.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.



## hci Command-Check-in

The hci command executes the check-in process specified on the command line or the first check-in process in the current context if no process name is specified. The process properties determine the default behavior for the process access and execution of linked processes. Other aspects of the process can be specified through command parameters read from the command line.

This command has the following format:

```
hci filename {-b name -en name -st name -p name -vp path} {-usr username -pw password}
[-v] [-pn name] [-ur] [-uk] [-ro] [-d] [-nd] [-de string] [-s filename | pattern] [-op
option] [-bo] [-ot | -ob] [-if option] [-dvp path] [-dcp path] [-cp path] [-rm name
-rport portnumber] [-rusr username] [-rpw password] [-prompt] [-i inputfile.txt | -di
inputfile.txt] [-eh filename] [-er filename] [-o filename | -oa filename] [-arg]
[-wts] [-tr] [-h]
```

### **filename**

Specifies the name of the file to check in.

### **-b name**

(Required) Specifies the CA Harvest SCM broker name.

### **-en name**

(Required) Defines the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

### **-st name**

(Required) Specifies the name of the state in which the user is operating.

### **-p name**

(Required) Specifies the CA Harvest SCM package name, read from the command line or a signature file.

### **-vp path**

(Required) Specifies the CA Harvest SCM view path that corresponds to the current working directory.

### **-usr username**

(Required, unless -prompt is specified) Defines the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

### **-pw password**

(Required, unless -prompt is specified) Defines the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-ur**

(Optional) Specifies that the item in the view is no longer reserved by the specified package after it is updated or created. This package cannot make any more changes until the item is checked out again. The file permission is set to read-only when this mode of check-in is used.

**-uk**

(Optional) Specifies that the item in the view is updated or created and the current package keeps it reserved so that more changes can be made.

**-ro**

(Optional) Specifies that no check-in is performed. The item is not updated, but the item is no longer marked as reserved for the current package. The file permission is set to read-only when this mode of check-in is used.

**-d**

(Optional) Specifies that files be deleted from the client directory after they have been successfully checked in.

**-nd**

(Optional) Specifies to not delete files from the client directory after they have been checked in successfully.

**-de *string***

(Optional) Specifies a comment to be associated with the new versions of items created during the process execution. If this option is not specified, you are prompted to enter a description. You can enter multiple lines of description in prompt mode.

**UNIX or Linux:** To terminate the description, enter a period (.) and a carriage return on an otherwise blank line, or use an end of file marker Ctrl+D.

**Windows:** To terminate the description, enter a period (.) and a carriage return on an otherwise blank line, or use an end of file marker Ctrl+Z on a blank line.

**-s filename | pattern**

(Optional) Specifies that the check-in process should search every directory below the current one for files that match a file name or naming pattern. This option must be followed by a file name or naming pattern. To specify multiple file names or naming patterns, use the -s option before the first file name or pattern. -s will search for the patterns specified between it and the next option.

Naming patterns that include a wildcard must be enclosed in single or double quotes when used with the -s option. If no quotes are included on multiple naming patterns, hci searches recursively for the first file (the naming pattern immediately following the -s option) and searches only the current directory for the remaining files.

**Examples: -s option**

To recursively check in all files with the .c file extension, use the following command:

```
hci [options] -s "*.c"
```

To check in file.txt, one.h, two.h, three.h and all .c files, plus all .exe files recursively use the following command:

```
hci file.txt one.h two.h three.h *.c [options] -s *.exe
```

**-op option**

(Optional) Specifies where files or items should be placed in relationship to the specified view path or client directory structure.

Specify one of the following options:

**p**

Preserve directory structure-Checks in the selected files to repository view paths with names that correspond to their client directory location, if these view paths currently exist.

**pc**

Preserve and create path structure-Checks in selected files to paths with names that correspond to their client directory location, and creates any view paths that do not currently exist.

**as**

All files to same view path-Checks in all selected files to the same path in the destination view, ignoring the client directory structure.

**-bo**

(Optional) Specifies that only the user who checked out an item is allowed to release the reserved version. This user is the owner of the lock or reservation. Setting this option at check-in (hci) prevents a user from releasing an item if that user did not initially reserve it.

**-if option**

(Optional) Specifies the item filter for check-in. Specify one of the following options:

**ne**

**New or Existing Items**-When this option is selected, all selected files are checked in if they are reserved by the package or did not previously exist in the repository. This filter does not require the corresponding item to be in the current view. However, if it is in the current view, then it must have a reserved version in the current package.

**no**

**New Items Only**-This option limits the check-in to files that do not have corresponding items in the current view. If the item has been removed from the current view, then the file can be checked in using this filter.

If the corresponding item was renamed in the current project, then the file cannot be checked in. The file must be renamed before it can be checked in.

**eo**

**Existing Items Only**-This option limits the check-in to files that have corresponding items reserved by the package. Any files without corresponding items are skipped. This filter can be used to prevent the existence of unwanted files, such as temporary files or templates, in your repository.

**-ot**

Limits the check-in to files that do not have corresponding items in the current view and places them on the project trunk.

**-ob**

Limits the check-in to files that do not have corresponding items in the current view and places them on a branch.

**Note:** The -ot (New Items on Trunk) and -ob (New Items on Branch) options are disabled if the user selects Existing Items Only; otherwise, one or both can be selected. If the user only selects New Items on Trunk, then new items can only be checked in to the trunk. If the user only selects New Items On Branch, then new items can only be checked in to the branch. If the user selects both, then new items can be checked in to either the trunk or the branch.

**-dvp path**

(Optional) Specifies a default view path.

**-dcp path**

(Optional) Specifies a default client path.

**-cp path**

(Optional) Specifies the client path that corresponds to current view path. The client path must be an absolute path. For items with reserved only tags, this option is required.

**-rm *name***

(Optional) Specifies the remote computer to check out from or check in to. Specify this option only if you want to check out from or check in to a remote computer.

**-rport *portnumber***

(Optional) Specifies the port number of the remote computer to check out from or check in to. Specify this option only if you want to check out from or check in to a remote computer.

**-rusr *username***

(Optional) Specifies the user name for the remote computer to check out from or check in to. Necessary only if you want to check out from or check in to a remote computer.

**-rpw *password***

(Optional) Specifies the password for the remote computer to check out from or check in to. Necessary only if you want to check out from or check in to a remote computer.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-er *filename***

(Optional) Specifies the encrypted remote username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility. -er overrides -musr and -mpw options.

**-tr**

(Optional) Turns on the local agent network activity trace.

## hcmpview Command-Compare View

The hcmpview command executes the first compare view process in the first project given. The process properties determine the default behavior for the process access and executing of linked processes. Other aspects of the process can be specified through command parameters read from the command line.

This command has the following format:

```
hcmpview {-b name -en1 name -en2 name -st1 name -vn1 name -vn2 name -vp1 path -vp2  
path} -usr username -pw password} [-uv1] [-uv2] [-cidc] [-ciic] [-s ] [-v] [-prompt]  
-i inputfile.txt | -di inputfile.txt] [-eh filename] [-o filename | -oa filename]  
[-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en1 *name***

(Required) Specifies the first CA Harvest SCM project context being compared. If the project name contains spaces, enclose the name in quotation marks.

**-en2 *name***

(Required) Specifies the second CA Harvest SCM project context being compared. If the project name contains spaces, enclose the name in quotation marks.

**-st1 *name***

(Required) Specifies the state context for the compare view process for the first project (-en1).

**-vn1 *name***

(Required) Specifies the name of the first view to be compared.

**-vn2 *name***

(Required) Specifies the name of the second view to be compared.

**-vp1 *path***

(Required) Specifies the first CA Harvest SCM view path to be compared.

**-vp2 *path***

(Required) Specifies the second CA Harvest SCM view path to be compared.

**-usr *username***

(Required, unless -prompt is specified) Defines the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Defines the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-uv1**

(Optional) Specifies items that are only in the specified view, but not in the other view. Both unique view1 and unique view2 can be selected at the same time.

**-uv2**

(Optional) Specifies items that are only in the specified view, but not in the other view. Both unique view1 and unique view2 can be selected at the same time.

**-cidc**

(Optional) Specifies that all items which are common to view1 and view2 but have different contents should be listed.

**-ciic**

(Optional) Specifies that all items which are common to view1 and view2 and have identical contents should be listed.

**-s**

(Optional) Specifies that the compare view process should include directories below the current one in the comparison.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hco Command-Check-out

The hco command executes the check-out process specified on the command line or the first check-out process in the current context if no process name is specified. The process definition determines the default behavior for the Replace Read-Only Files option, process access, and execution of linked processes. Other aspects of the process can be specified through command parameters read from the command line.

The options -up, -br, -ro, -sy, and -cu are for Update, Browse, Reserve Only, Synchronize, and Concurrent Update, respectively.

This command has the following format:

```
hco filename(s) | pattern {-b name -en name -st name -vp path -p name}{-up | -br |  
-ro | -sy | -cu} {-usr username -pw password} [-vn] [-nvs] [-nvf] [-r] [-replace option]  
[-v] [-nt] [-ss name] [-s filename | pattern] [-pf name [-po]] [-bo -to -tb] [-ced]  
[-dvp path] [-dcp path] [-cp path] [-op option] [-pn name] [-rm name -rport portnumber]  
[-rusr username] [-rpw password] [-prompt] [-i inputfile.txt | -di inputfile.txt]  
[-eh filename] [-er filename] [-o filename | -oa filename] [-arg] [-wts] [-tr] [-h]
```

**filename(s) | pattern**

Specifies file name(s) or naming pattern used for check-out.

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-en name**

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st name**

(Required) Specifies the name of the state in which the user is operating.

**-vp path**

(Required) Specifies the CA Harvest SCM view path that corresponds to the current working directory.

**-p name**

(Required, unless -br or -sy is specified.) Specifies the package context to be used for update, concurrent update, and reserve only modes, read from the command line or a signature file.

**Note:** Package context is ignored for check-out for browse and synchronize.

**-up**

(Required: -up, -br, and -ro are mutually exclusive and one is required.) Specifies that the items should be copied to the destination directory and a reserved version for each created, allowing the files to be checked back in. Only the latest trunk version of an item, or the latest version on an unmerged branch, can be checked out for Update.

**-br**

(Required: -up, -br, -ro, -sy, and -cu are mutually exclusive and one is required.) Specifies that the items should be copied to the destination directory but does not allow you to check the files back in. The file permission is set to read-only. If no version number (-vn) or branch only (-bo) options are specified, Browse mode selects the latest trunk version.



**-ro**

(Required: -up, -br, -ro, -sy, and -cu are mutually exclusive and one is required.) Specifies that no data is to be moved to external directories, but the items are marked as reserved so they can be checked back in.

**-sy**

(Required: -up, -br, -ro, -sy, and -cu are mutually exclusive and one is required.) Specifies synchronize mode, which identifies the versions of the files in the client file system by using the signature file. CA Harvest SCM versions are checked out only if the signature file shows the following differences:

- Client files that differ from their corresponding CA Harvest SCM versions.
- Client files whose timestamps differ from their corresponding timestamps in the signature file.
- Items are checked out in read-only mode. The check out for synchronize mode is especially useful in the build process.

**Note:** For more information on signature files, see the *Workbench User Guide*.

**-cu**

(Required: -up, -br, -ro, -sy, and -cu are mutually exclusive and one is required.) Specifies that the items be copied to the destination directory and a reserved version created for each, allowing the files to be checked back in. Unlike Update, Concurrent Update creates the reserved version on a branch of the trunk. The permission on a read-only file is changed to normal (writable) when this mode of check-out is used.

**-usr *username***

(Required, unless -prompt is specified) Defines the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Defines the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-vn**

(Optional) Specifies a version of an item for browse or concurrent update. If no version is specified with the -vn option, the latest version of the item in the view associated with the specified state is checked out.

Occasionally, this algorithm can lead to errors. If, for example, you are trying to check out an item for update and later versions exist in another view, hco will generate an error. To check out such an item, you must use the Concurrent Update mode.

Specifying more than one -vn parameter on the command line check-out is not allowed. After a version is specified with the -vn option, that version is used for all files checked out.

**Note:** The -vn option cannot be used with the Synchronize (-sy) or Update (-up) modes of check-out or update.

**-nvs**

(Optional) Specifies that latest, normal, and trunk versions are recursively selected for check-out.

When -nvs is used, the [version] system variable will not be populated in processes linked to the check-out process.

This option is intended to decrease the memory requirements for checking out a large amount of versions.

**Note:** The -nvs option cannot be used with options: -po, -pf, -bo, -vn, -s, -tb, or -dvp.

**-nvf**

(Optional) Specifies how CA Harvest SCM handles “file not found” errors that occur when the hco command is running.

When you specify -nvf, if one or more specified items is missing or if no search results found, CA Harvest SCM does the following actions:

- Aborts the check out
- Displays an error message
- Returns error code 14

When you do not specify -nvf, if one of the conditions occurs, then CA Harvest SCM continues the check out, displays a warning message, and returns a “successful” code.

**-r**

(Optional) Specifies whether the checked-out files should replace existing read-only files on the host. When -r is specified, existing read-only files are replaced. This lets you replace files that you previously checked out for Browse, or checked in complete, without having to manually modify the file's access. If you do not specify the -r option, the check out does not replace any existing read-only files, regardless of the Administrator application setting.

Files that exist with write permission are never overwritten. CA Harvest SCM assumes that such a file has been checked out for Update and not checked back in yet. Overwriting such a file might cause a user to lose unsaved changes.

If no corresponding file exists, the item is checked out regardless of whether the `-r` option is used.

**UNIX and Linux:** The Replace Read-Only Files option replaces files only if the user who is executing the check-out process owns them.

**-replace**

(Optional) Specifies if existing files are to be replaced during check-out. Specify one of the following options:

**ro**

Replaces read-only files only.

**rw**

Replaces writable (read/write) files only.

**all**

Replaces both read-only and writable files.

**Note:** `-replace` and `-r` are mutually exclusive.

**-nt**

(Optional) Specifies versions with normal tag.

**-ss *name***

(Optional) Specifies the name of the snapshot view. If used in the check-out process, the state specified must have All Snapshot Views enabled.

**-s *filename* | *pattern***

(Optional) Specifies that the check-out process should search every directory below the current one for files that match a file name or naming pattern. This option must be followed by a file name or naming pattern. To specify multiple file names or naming patterns, use the `-s` option before the first file name or pattern. `-s` will search for the patterns specified between it and the next option.

Naming patterns that include a wildcard must be enclosed in single or double quotes when used with the `-s` option. If no quotes are included on multiple naming patterns, `hco` searches recursively for the first file (the naming pattern immediately following the `-s` option) and searches only the current directory for the remaining files.

### Examples: -s option

To recursively check out all files with the .c file extension, use the following command:

```
hco [options] -s "*.c"
```

To check out \*.c and \*.exe and recursively check out the files one.c, two.c, and three.c, use the following command:

```
hco file.txt *.c ... -s one.c two.c three.c [option] *.exe
```

### **-pf**

(Optional) Specifies that items should be selected based on package contents. Checks out the latest versions from the current view of all items that have any version associated with the specified package. If used with the -po option, checks out the latest versions associated with the -pf specified package. Specifying more than one -pf parameter is not allowed.

### **-po**

(Optional) Specifies that the latest version should be checked out from the list of packages passed in by the -pf option. -po is a sub-option of -pf and cannot be used by itself.

### **-bo**

(Optional) Specifies to check out only versions located on branches. If you specify a trunk version using the -vn option and also specify -bo, no version will be selected.

### **-to**

(Optional) Specifies to check out only versions located on the trunk. This is the default. If you specify a branch version using the -vn option and also specify -to, no version will be selected.

### **-tb**

(Optional) Specifies to check out versions located on the trunk and branches.

### **-ced**

(Optional) Enables the creation of directories that contain no items during a recursive check-out. The default is to create no empty directories.

### **-dvp path**

(Optional) Specifies a default view path.

### **-dcp path**

(Optional) Specifies a default client path.

### **-cp path**

(Optional) Specifies the client path that corresponds to current view path. The client path must be an absolute path.

**-op *option***

(Optional) Specifies where files or items should be placed in relationship to the specified view path or client directory structure. Specify one of the following options:

**p**

**Preserve directory structure**-Checks out all selected items into corresponding client directories, if they already exist. If the directories do not exist, an error message is displayed and the items are not checked out.

**pc**

**Preserve and create path structure**-Checks out selected files to paths with names that correspond to their client directory location, and creates any view paths that do not currently exist.

**as**

**All files to same view path**-Checks out all selected files to the same path in the destination view, ignoring the client directory structure.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type for which the user has execute access and that allows the specified mode in the specified state is invoked.

**-rm *name***

(Optional) Specifies the remote computer to check out from or check in to. Specify this option only if you want to check out from or check in to a remote computer.

**-rport *portnumber***

(Optional) Specifies the port number of the remote computer to check out from or check in to. Specify this option only if you want to check out from or check in to a remote computer.

**-rusr *username***

(Optional) Specifies the username for the remote computer to check out from or check in to. Specify this option only if you want to check out from or check in to a remote computer.

**-rpw *password***

(Optional) Specifies the password for the remote computer to check out from or check in to. Specify this option only if you want to check out from or check in to a remote computer.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-er**

(Optional) Specifies the encrypted remote username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility. -er overrides -rusr and -rpw options.

**-tr**

(Optional) Turns on the local agent network activity trace.

**More information:**

[Return Codes](#) (see page 173)

[Common Cases](#) (see page 174)

[Return Code Examples](#) (see page 175)

## hcp Command-Create Package

The hcp command executes the create package process specified on the command line, or the first create package process in the current context if no process name is specified. The process definition determines the default behavior for associated forms and the state of the package, as well as process access and execution of linked processes. Other aspects of the process can be specified through command parameters read from the command line.

This command has the following format:

```
hcp packagename {-b name -en name -st name -usr username -pw password} [-v] [-pn name]
[-at username] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename]
[-o filename | -oa filename] [-arg] [-wts] [-h]
```

***packagename***

Specifies the name of the package to create.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *name***

(Required) Specifies the name of the state in which the user is operating.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-at *username***

(Optional) Specifies the user name to assign to the package.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hcpj Command-Copy Project

The hcpj command copies a current project to a new project or switches a project between active and inactive. To toggle project status only, do not include the new name flag. Toggle can be active or inactive only. If the name of the new project exists, then hcpj uses the default project name. The hcpj command executes with the default Duplicate Access Control option, unless you specify the -dac option to clear it. You can select the Create User Group option by specifying the -cug option.

Projects cannot be copied to the same folder in which they are located. Also, projects located in Active Projects or Inactive Projects can be copied only to the Lifecycle Templates folder. The -cpj option is not required and may have one of these mutual exclusive values: -act, -ina, or -tem. The -act and -ina options should be used only for projects located in the Lifecycle Templates folder to indicate that the project should be copied to the Active Projects folder (-act) or copied to the Inactive Projects folder (-ina). The -tem option should be used only for projects located in the Active Projects or Inactive Projects folder.

This command has the following format:

```
hcpj {-b name -cpj name} [-npj name] {-act | -ina | -tem} {-usr username -pw password}  
[-dac] [-cug] [-v] [-prompt] [-i inputfile.txt | -di inputfile.txt]  
[-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-cpj *name***

(Required) Specifies the source project to copy.

**-npj *name***

(Optional) Specifies the new name for the project you are copying.

**-act**

(Required: -act, -ina, and -tem are mutually exclusive and one is required.) Specifies that the project you are copying be set to active status.

**-ina**

(Required: -act, -ina, and -tem are mutually exclusive and one is required.) Specifies that the project you are copying be set to inactive status.

**-tem**

(Required: -act, -ina, and -tem are mutually exclusive and one is required.) Specifies that the project you are copying is to be a template.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-dac**

(Optional) Specifies not to duplicate access control. By default, when the -dac option is not specified, access control is duplicated.

**-cug**

(Optional) Specifies to create a user group with the same name as the project and also grant View and Use access to this user group.



**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hcropmrg Command-Cross Project Merge

The hcropmrg command executes the first cross project merge process in the first project given. The process properties determine the default behavior for the process access and executing of linked processes. Other aspects of the process can be specified through command parameters read from the command line.

This command has the following format:

```
hcropmrg {-b name -en1 name -en2 name -st1 name -st2 name -p1 name -p2 name1 name2}  
{-usr username -pw password} [-pn] [-mo number] [-plo number] [-vfs snapshotname]  
[-mb] [-v] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename] [-o  
filename | -oa filename] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en1 *name***

(Required) Specifies the destination CA Harvest SCM project. If the project name contains spaces, the name should be enclosed by quotation marks.

**-en2 *name***

(Required) Specifies the source CA Harvest SCM project. If the project name contains spaces, the name should be enclosed by quotation marks.

**-st1 *name***

(Required) Specifies the destination CA Harvest SCM state.

**-st2 *name***

(Required) Specifies the source CA Harvest SCM state.

**-p1 *name***

(Required) Specifies the destination CA Harvest SCM package name.

**-p2 *name1 name2***

(Required) Specifies the source CA Harvest SCM package name or names.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-mo *number***

(Optional) Specifies the merge option.

**Default:** 1

Specify one of the following options:

**1**

**Merge Conservatively**-Creates a merge-tagged version, regardless of the contents of the versions. An exception is when the branch version is the latest version in the view; in this case, it is closed and a normal version is created.

**2**

**Merge Aggressively**-Creates a merge-tagged version only when conflicts are found. If no conflicts are found, the branch and trunk versions are merged to create a normal version.

**Note:** A conflict occurs when a set of lines is modified in both the branch and the trunk; insertions and deletions are not conflicts.

**3**

**Take Trunk (Target) Version**-Automatically selects the trunk (target) to create the final version, without comparing the contents of the versions. Selecting this option creates a normal version on the trunk and closes the branch.

**4**

**Take Branch (Source) Version**-Automatically selects the branch (source) to create the final version, without comparing the contents of the versions. Choosing this option creates a normal version on the trunk and closes the branch.

**-plo *number***

(Optional) Specifies the target location of the merge version. The placement options let you specify merge version destinations.

**Default:** 3

Specify one of the following options:

**1**

**Branch Only**-Creates a merge version on the target branch. This allows changes to be copied from the source project to the target project even if one or more target items are reserved for update in the main trunk. With this option, a branch is created to store the changes.

**2**

**Trunk Only**-Creates a merge version on the target trunk.

**3**

**Trunk or Branch**-Creates a merge version on the target trunk or branch. This allows changes to be copied from the source project to the target project even if one or more target items are reserved for update in the main trunk. If items are reserved for update on the trunk, a branch is created to store the changes. If items are not reserved for update on the trunk, the items are simply copied to the trunk.

**-vfs**

Specifies the snapshot name from which the versions need to be merged.

Ex:- hcropmrg -b brokername -en1 "target project" -en2 "Source Project" -st1 target state -p1 "target package" -vfs "snapshotname1" -eh "credentials" -mo 2 -plo 3

**Note:** -vfs and -st2 are mutually exclusive options. Similarly -vfs and p2 options are mutually exclusive.

**-mb**

Specifies if the cross project merge process must use the base versions also in the snapshot. By default, the base versions are not processed for the merge process. This option can be used only with -vfs option

Ex:- hcropmrg -b brokername -en1 " target project " -en2 " Source Project " -st1 "target state" -p1 "target package" -vfs "snapshotname1" -eh "credentials" -mo 2 -plo 3 -mb

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hcrulte Command-Correlate Package

The hcrulte command correlates the CA Harvest SCM package identification specified on the command line to the Enterprise package identification of CA Change Manager Enterprise Workbench (CA CMEW).

```
hcrulte CA_SCM_package_ID {-b name -en name} -usr username -pw password {-epid Enterprise_package_ID -epname Enterprise_package_name} [-v] [stat status] [-prompt] [-eh filename] [-o filename]
```

You can uncorrelate a CA Harvest SCM package from a CA CMEW package by using the -d option, for example:

```
hcrulte CA_SCM_package_ID -b broker -usr username -pw password -epid Enterprise_package_ID -d
```

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-en name**

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-usr username**

(Optional) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw password**

(Optional) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-epid Enterprise\_package\_ID**

(Required) Specifies the Enterprise package identification of CA CMEW.

**-epname Enterprise\_package\_name**

(Required) Specifies the Enterprise package name of CA CMEW.

**-d**

(Optional) Uncorrelates the CA Harvest SCM package specified on the command line from the CA CMEW package.

**-eh filename**

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hcrtpath Command-Create Empty Path

The hcrtpath command allows you to create an empty view path. The empty path created by hcrtpath is visible to all projects. The check-in process you provide (specified with -cipn) is used by hcrtpath to check for access rights. You must have Execute access on that check-in process and the check-in process must have New or New or Existing item filters enabled in order for the hcrtpath to create an empty path without errors. If you do not specify a check-in process, the first valid check-in process in the current state is used.

This command has the following format:

```
hcrtpath {-b name -en name -st name -rp path} {-usr username -pw password} {-p name}  
[-cipn name] [-de description] [-ot | -ob] [-v] [-prompt] [-i inputfile.txt | -di  
inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the broker name.

**-en *name***

(Required) Specifies the project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *name***

(Required) Specifies the name of the state in which the user is operating.

**-rp *name***

(Required) Specifies the repository path name to be created. This path name value is compared to the repository to determine what new sub item path or paths are to be created. For example if the -rp value is "\Rep1\DirA\DirB\DirC" and the current view in the repository is "\Rep1\DirA", the view paths "DirB\DirC" will be created under "\Rep1\DirA".

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-p *name***

(Required) Specifies the package name, read from the command line or a signature file.

**-cipn *name***

(Optional) Specifies the name of the particular check-in process to be invoked, if more than one check-in process of the same type exists in the state. If no check-in process name is included in the command, the first check-in process of that type in that state is invoked.

**-de *description***

(Optional) Specifies a comment to be associated with the new path created during the process execution. You can enter multiple lines of description in the prompt mode.

UNIX or Linux: To terminate the description, enter a period (.) and a carriage return on an otherwise blank line, or use an end of file marker Ctrl+D.

Windows: To terminate the description, enter a period (.) and a carriage return on an otherwise blank line, or use an end of file marker Ctrl+Z on a blank line.

**-ot**

(Optional) Specifies that the new path version is to be created on the trunk. This parameter is mutually exclusive with **-ob**.

**Note:** This option is only available if the selected or default check-in process allows this operation to be applied on the trunk.

**-ob**

(Optional) Specifies that the new path version is to be created on a branch. This parameter is mutually exclusive with **-ot**.

**Note:** This option is only available if the selected or default check-in process allows this operation to be applied on a branch.

**-eh *filename***

(Optional) Specifies the encrypted username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hdbgctrl Command-Debug Control

The broker `-qtrace` option, the hserver `-logging` option, and the remote agent `-trlvl` option can be turned on or off at runtime by using the hdbgctrl debug control utility. This utility is an administration tool, so the user must have the CA Harvest SCM administrator privileges to run this command. The trace messages are written to the default process log file.

This command has the following format:

```
hdbgctrl [-b name] [-rm name -rport portnumber] [-usr username] [-pw password]  
[-prompt] [-eh filename] [-o filename | -oa filename] [-h] debug_control_string
```

**-b *name***

(Required, must specify `-b` or `-rm`) Specifies the CA Harvest SCM broker name. This option is required if the target process is the broker or the hserver.

**-rm *name***

(Required, must specify `-b` or `-rm`) Specifies the remote agent computer name. This option is required if the target process is the agent.

**-rport *portnumber***

(Required, must specify `-b` or `-rm`) Specifies the port number of the remote agent computer. This option is required if the agent computer name is specified.

**-usr *username***

(Required, unless `-prompt` is specified) Specifies the user name for login to the broker or agent. If `-usr` is defined without `-pw`, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless `-prompt` is specified) Specifies the password for login to the broker or agent. To hide the password from being exposed in plain text on the command line, use the `prompt` option.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM or agent username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the `svrenc` utility.

**Note:** `-eh` overrides `-usr` and `-pw` options. `-prompt` overrides `-eh`, `-usr` and `-pw` options.

### Debug\_control\_string

Controls the trace options for the broker, the hserver, and the remote agent. The string has the following format:

process\_name:switch:logging\_level:transaction\_name:client\_host

#### process\_name:broker, hserver, or agent

Specifies which process is controlled. This field is required for all target processes.

#### switch: on or off

Specifies enable (on) or disable (off) the trace for the target process. This field is required for all target processes.

#### logging\_level: 1-5 for hserver, 1-3 for agent

Specifies the level of the trace (-logging for hserver and -trlvl for the agent) for the target process. This field is required if the target process is the hserver or the agent and the switch is on. This field is ignored if the target process is the broker or the switch is off.

#### transaction\_name: hserver transaction name or "all" for all transaction types

Specifies which transaction the hserver trace is activated. The trace is turned off automatically at the end of the transaction. This field is required for the hserver. The field is ignored for the broker and the agent. Only one transaction type can be activated for debug trace at any time unless "all" is specified for the transaction\_name.

#### client\_host: client\_machine\_name

Activates the hserver trace only if the transaction request is from the specified client and the transaction matches the specified transaction name. This field is optional for the hserver and is ignored if the target process is the broker or the agent.

### Examples: hdbgctrl Command

```
hdbgctrl -b scmbroker -usr scmuser -pw passwd broker:on
hdbgctrl -b scmbroker -usr scmuser -pw passwd broker:off
hdbgctrl -b scmbroker -usr scmuser -pw passwd hserver:on:4: LIST_VERSION_IN_VIEW
hdbgctrl -b scmbroker -usr scmuser -pw passwd hserver:on:4:
LIST_VERSION_IN_VIEW:machine_A
hdbgctrl -b scmbroker -usr scmuser -pw passwd hserver:off:4: LIST_VERSION_IN_VIEW
hdbgctrl -rport portnumber -rm scmagent -usr scm agent username -pw scm agent password
agent:on:1
hdbgctrl -rport portnumber -rm scmagent -usr scm agent username -pw scm agent password
agent:off:1
```



## hdelss Command-Delete Snapshot View

The hdelss command deletes snapshot views specified on the command line. Snapshot views are specified by names. Snapshot view names are unique per project but not across projects; therefore, you must supply the project name with the `-en` option. The delete snapshot view command has the same access control as the Administrator interface for deleting a snapshot view; that is, you must have at least one of the following access privileges to execute this command-line utility successfully:

- administrator (super user)
- admin project (harvest level of admin project access)
- update project (the project contains this snapshot view)

This command has the following format:

```
hdelss snapshotview_name {-b name} {-en name} [-usr username] [pw password] [-prompt]
[-eh filename] [o filename | -oa filename] [-v] [-wts] [-h]
```

### **snapshotview\_name**

(Required) Specifies one or more snapshot view names that you want to delete.

### **-b name**

(Required) Specifies the CA Harvest SCM broker name.

### **-en name**

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

### **-usr username**

(Required, unless `-prompt` is specified) Specifies the user name for login to the broker. If `-usr` is defined without `-pw`, the user is prompted for the password at runtime.

### **-pw password**

(Required, unless `-prompt` is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the `-prompt` option.

### **-eh filename**

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the `svrenc` utility.

**Note:** `-eh` overrides `-usr` and `-pw` options. `-prompt` overrides `-eh`, `-usr` and `-pw` options.

## hdlp Command-Delete Packages

The hdlp command deletes the packages specified on the command line. Packages are specified by names. Package names are unique per project but not across projects; therefore, the project name must also be supplied with the -en option. A delete package process must exist in the state containing the package and the user must have permission to execute this process in order to delete a package. You can use the -pn option to specify the delete package process to use for the deletion.

This command has the following format:

```
hdlp {-b name} {-en name} {-pkgs name...} [-st name] [-pn process] [-usr username]
[-pw password] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename] [-o
filename | -oa filename] [-v] [-arg] [-wts] [-h]
```

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-en name**

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-pkgs name...**

(Required) Specifies one or more package names.

**Limit:** A maximum of 50 packages names can be supplied.

**-st name**

(Optional) Specifies the name of the state in which the user is operating.

**-pn name**

(Optional) Specifies the name of the delete package process for which the user has execute privileges.

**-usr username**

(Optional) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw password**

(Optional) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-eh filename**

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hdp Command-Demote Package

The hdp command executes the demote process specified on the command line, or the first demote process in the current context if no process name is specified. The process definition determines the default behavior, process access, and execution of linked processes. Other aspects of the process can be specified through command parameters.

This command has the following format:

```
hdp package_name... {-b name -en name -st name} {-usr username -pw password} [-v] [-pn name] [-pb] [-pd] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-pdr] [-vdr] [-adp] [-h]
```

### **package\_name...**

Specifies the names of one or more packages to demote.

### **-b name**

(Required) Specifies the CA Harvest SCM broker name.

### **-en name**

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

### **-st name**

(Required) Specifies the name of the state in which the user is operating.

### **-usr username**

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

### **-pw password**

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

### **-pn name**

(Optional) Specifies a specific process name to invoke when more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

### **-pb**

(Optional) Specifies that all the packages that belong to a bound package group must be promoted together.

**-pd**

(Optional) Specifies that the packages which depend on other packages must not be demoted. This option verifies package dependency. Dependency is based on versions in the current view. In the current state, a package with a lower item-version cannot be demoted without also demoting the packages with the higher item-versions in the current view. If the demote process is in a state that shares the same view as the one in the Demote To state, then this option is not enforced.

**Note:** The higher item-version causes a dependency error when it is on the trunk or on the branch.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted the username-password credentials. Use the svrenc utility to generate this file name.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-pdr**

(Optional) Generates the package dependency report by listing all dependent packages. The Demote process is not executed when this option is specified.

**-vdr**

(Optional) Generates the package dependency report by listing all dependent versions. The Demote process is not executed when this option is specified.

**-adp**

(Optional) Adds dependent packages in the current state to the execution list, when specified.

## hdv Command-Delete Version

The hdv command executes the delete version process that is specified on the command line, or the first delete version process in the current context, if no process name is specified. The process definition determines the default behavior and the process access. Other aspects of the process can be specified through command parameters.

This command has the following format:

```
hdv filename {-b name -en name -st name -vp path} {-usr username -pw password} [-pn
name] [-v] [-p name] [-s version_name | pattern] [-prompt] [-i inputfile.txt
| -di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

***filename***

Specifies the file name of the file to delete.

***-b name***

(Required) Specifies the CA Harvest SCM broker name.

***-en name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

***-st name***

(Required) Specifies the name of the state in which the user is operating.

***-vp path***

(Required) Specifies the CA Harvest SCM view path that corresponds to the current working directory.

***-usr username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**Note:** If the user is not the administrator or the creator of the version, the respective versions get deleted or restricted, based on the delete\_by\_owner\_only process property.

***-pw password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

***-pn name***

(Optional) Specifies a specific process name to invoke, if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

***-s version\_name | pattern***

(Optional) Specifies that the delete version process search every directory below the current one for versions that match the naming pattern. This option is followed by a version name or wildcard pattern (for example, \*.c).

Naming patterns that include a wildcard must be enclosed in single or double quotes when used with the -s option.

***-p name***

(Required for deleting branch versions only) Specifies the package context that is associated with the version to be deleted.

### Example: -s option

To recursively delete all versions with the .c file extension, use the following command:

```
hdv [options] -s "*.c"
```

**Note:** If no quotation marks are included, the wildcard is expanded and a list of versions that are inserted into the hdv operation before the process is invoked. This option causes hdv to do a recursive search for the first version (the version name immediately following the -s option on the command line). But searches only the current directory for the remaining versions.

### -eh *filename*

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name that is used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hexecp Command-Execute Program

The hexecp command executes the program specified in by the -prg argument, on the computer specified by the -m argument, with options specified by the -args argument. Output is directed to standard output unless a file is named using the -o option.

**Note:** -usr and -pw are not your CA Harvest SCM username and password, but rather the username and password for the computer that the program is executed on.

hexecp attempts to log in to an agent. If it does not find an agent on the computer you have designated after the -m option, you receive a message stating, "An agent for *computername* cannot be found." Before executing hexecp, ensure that an agent is running on the target computer.

This command has the following format:

```
hexecp {-prg program -m computername -rport portnumber} {-syn | -asyn} {-usr username -pw password} [-args arguments...] [-ma] [-v] [-prompt] [-i inputfile.txt | di inputfile.txt] [-er filename] [-o filename | -oa filename] [-arg] [-wts] [ h]
```

### -prg *program*

(Required) Specifies the full path and program you want to execute on the destination computer (-m).

**-m *computername***

(Required) Specifies the name of the destination computer on which you want to execute the program.

**-rport *portnumber***

(Required) Specifies the port number of the destination computer on which you want to execute the program.

**-syn**

(Required: -syn and -asyn are mutually exclusive and one is required.) Specifies synchronous mode.

**-asyn**

(Required: -syn and -asyn are mutually exclusive and one is required.) Specifies asynchronous mode.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for the login to the computer that the program is executed on. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for the login to the computer that the program is executed on. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-args *arguments...***

(Optional) Specifies one or more arguments you want the program, specified by -prg, to execute on the destination computer. Each argument may or may not contain spaces. For example:

```
-args "a 1" "a 2" "a3" a4
```

If an argument contains a space, it is passed to the user's process as one argument with quotation marks. For example, "a 1" is passed to the user's process as "a 1".

If an argument does not contain a space, it is passed to the user's process without quotation marks. For example, "a3" and a4 are passed to the user's process as a3 and a4.

**-ma**

(Optional) Allows multiple arguments to be executed on the destination computer.

**-er *filename***

(Optional) Defines the encrypted remote username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility. -er overrides -musr and -mpw options.

## hexpenv Command-Export

The export utility lets you copy life cycle definitions from one CA Harvest SCM installation to another. This process copies all access data by default unless you specify the flag `-eac` to clear it. You can create a user group with the same name as the project by specifying the Create User Group (`-cug`) option.

After a successful export execution, one file is created in the export destination location. The `.har` file is a text file that stores the logical records.

This command has the following format:

```
hexpenv {-b name -en name -f filename} {-usr username -pw password} [-eac] [-cug]
[-eug] [-v] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename]
[-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-f *filename***

(Required) Specifies a name for the file that will receive the project files. The file name will automatically be appended with the `.har` file suffix.

**-usr *username***

(Required, unless `-prompt` is specified) Specifies the user name for login to the broker. If `-usr` is defined without `-pw`, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless `-prompt` is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the `-prompt` option.

**-eac**

(Optional) Specifies that export access control option should be cleared. If this option is not specified, hexpenv executes with the default export access control option selected.

**-cug**

(Optional) Specifies to create a user group with the same name as the project and grant Use access to that user group.



**-eug**

(Optional) Specifies to export a user group for the approve, notify, and linked notify processes to the newly created project.

**-eh filename**

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hfatt Command-Form Attachment

The hfatt command adds or removes form attachments or retrieves a file-type attachment. The hfatt command updates the attachment list for the specified form. If the attachment type is a Uniform Resource Locator (URL), then the attachment name is the URL. If the attachment type is a file, the name is the file name. A client path must be included to add a file-type attachment.

Do not specify the compress files option when adding files that are already in compressed format. Using the compress option on a file that is already compressed does not reduce the size of the attachment and can actually increase it. Compressed files usually have one of the following extensions: .cab, .gz, .zip, .jpg, .gif, .asf, .ram, .mp3, .wav.

This command has the following format:

```
hfatt {-b name} {-fn name} {-fid id} {-add} {-rem} {-get} [-at type] {-usr username  
-pw password} {-ft formtype name} [-comp] [-prompt] [-i inputfile.txt | -di  
inputfile.txt] [-v] [-cp path] [-rm computername -rport portnumber] [-rusr username]  
[-rpw password] [-eh filename] [-er filename] [-o filename | -oa filename] [-arg] [  
wts] [-h] form_attachment_name
```

The following rules apply to using the form name (-fn), form type (-ft), and form ID (-fid) options:

- Form must be specified by either form name (-fn) or object id (-fid).
- If you specify form name (-fn), then you must specify form type (-ft) and do not specify form ID (-fid).
- If you specify form ID (-fid), then do not specify form name (-fn) and form type (-ft).

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-fn *name***

(Required, unless -fid is specified.) Specifies the form name.

**-fid *id***

(Required, unless -fn is specified.) Specifies the form ID.

**-add**

(Required, unless -rem or -get is specified.) Specifies to add an attachment.

**-rem**

(Required, unless -add or -get is specified.) Specifies to remove the attachment.

**-get**

(Required, unless -add or -rem is specified.) Specifies to get the attachment.

**-at *type***

(Optional) Specifies the attachment type:

**F**

Specifies that the attachment is a file.

**U**

Specifies that the attachment is a URL.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-ft *formtype name***

(Required, unless -fid is specified.) Specifies the form type, and is required only if you are adding a form attachment by name.

**-comp**

(Optional) Specifies to compress files before adding as attachments.

**-cp *path***

(Optional) Specifies the client path that corresponds to current view path. The client path must be an absolute path.

**-rm *computername***

(Optional) Specifies the remote computer from which to retrieve the attachment. Necessary only if you want to retrieve from a remote computer.

**-rport *portnumber***

(Optional) Specifies the port number of the remote computer from which to retrieve the attachment. Necessary only if you want to retrieve from a remote computer.

**-rusr *username***

(Optional) Specifies the user name for the remote computer from which to retrieve the attachment. Necessary only if you want to retrieve from a remote computer.

**-rpw *password***

(Optional) Specifies the password for the remote computer from which to retrieve the attachment. Necessary only if you want to retrieve from a remote computer.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-er *filename***

(Optional) Specifies the encrypted remote username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility. -er overrides -musr and -mpw options.

## hformsync Command-Update Form Template

The hformsync command saves or updates XML form templates for a specified set of form types.

This command has the following format:

```
hformsync {-b name} {-usr username -pw password} [-prompt] [-d folder] [-hfd] [-f]
[-i inputfile.txt | -di inputfile.txt] [-v] [-eh filename] [-o filename | -oa filename]
[-arg] [-wts] [-h] [ -all ] [ form_type_name [... form_type_name ] ]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-all**

(Optional) Process all the form type definitions in the form reference folder rather than processing form types by name.

**-f**

(Optional) Do not check the relative timestamps of the file template in the reference folder and the corresponding CA Harvest SCM database version. Store the reference folder version to the CA Harvest SCM database. By default, the reference folder and database version of the template are compared and the database version is updated only if the reference folder version has a more recent timestamp.

**-d *directory***

(Optional) Specifies the form reference folder to be used for the command. By default, the form templates are converted/processed from the folder %CA\_SCM\_HOME%/forms.

**-hfd**

(Optional) Specifies that the form type definition file (HFD), if found should be used to create the form type XML template prior to storing the XML template to the database. By default, this option, no HFD conversion is performed.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted usernamepassword credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hft Command-Footprint

The hft command performs keyword substitution and is the basis for footprinting. The hft command must be executed from the location where the signature file and the input file (infile) reside. When hft executes, it reads the file specified for infile and searches for the pattern @(#) on each line. If this pattern is detected on a line, all keywords that represent signature attributes are substituted with the current values for the infile found in the signature file. The output is written to the file specified with -fo option. If this option is not used, the output is written back to the infile. The hft command can only be executed successfully from the location where the signature file and the infile reside.

**Note:** The infile should be checked in prior to running the hft command line utility. Checking in the infile will add information used by hft to the signature file. If an infile has not been checked in to CA Harvest SCM, when hft is run using that infile an hft.log file will return the warning message "hft warning: there is no signature file information for 'infile.txt'. success... hft has been executed successfully."

This command has the following format:

```
hft infile [-fo filename] [-a attributes...] [-b] [-fs] [-i inputfile.txt  
| -di inputfile.txt] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

Valid keywords are: version, versionid, tag, package, environment, state, viewpath, user, author, crtime, modtime, size, status, and item. All keywords inside *infile* must be enclosed by brackets, for example, [state].

### ***infile***

Specifies the file name of the input file containing the keyword substitutions.

### ***-fo filename***

(Optional) Specifies the name of a file to contain the output of the command execution. If not specified, the infile is overwritten by the output.

### ***-a attribute...***

(Optional) Specifies values for additional user-defined attributes not present in the signature file. Any string can be substituted, as long as you use the correct syntax and provide a value for it on the command line. However, the -a option cannot be used to override attributes defined in the signature file.

**-b**

(Optional) Specifies binary mode. Typically, hft will replace any marker of format “[tag]” with its corresponding value. If the value is longer than the marker itself, the file is expanded. For binary files, expanding the file corrupts it. To avoid corruption, specify -b, so that hft still replaces the marker with its corresponding value, but the value is truncated if it is longer than the marker.

**Example: -b option**

If the state is “Development”:

Infile: @(#) the state name is [state].

HFT **without** -b mode will produce an expanded line:

@(#) the state name is Development.

HFT **with** -b will produce the same line length:

[state                    ]

Then, hft with the -b option will produce:

@(#) the state name is De

@(#) the state name is Develo.

The state name is truncated in order to avoid expanding the file and corrupting the (binary) file structure. To see the full file without truncation, hft supports long tags.

Using the previous example, you can make the [state] tag longer:

@(#) the state name isdevelopment                    .

**-fs**

(Optional) Specifies to format the output view path information with forward slashes.

## hgetusg Command-Get Usage

The hgetusg command is used to retrieve the concurrent usage information from the broker. The information that can be retrieved from the broker includes peak concurrent usage and current usage. The peak concurrent usage information contains the maximum number of concurrent users registered with the broker during the period after the broker started and before the broker received the request. The current usage information contains the number of concurrent users registered with the broker at the time the broker received the request. The usage information includes the date and time of the report, the number of licensed users, the total number of users, user name, client URL, and client register time, and also includes the date and time of the peak usage.

**Note:** Both of the options, -pu and -cu, can be specified at the same time but at least one of them must be specified. To run hgetusg, you must log in as a CA Harvest SCM administrator.

This command has the following format:

```
hgetusg {-b name} {-usr username -pw password} [-pu] [-cu] [-prompt] [-o filename | -oa filename] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-pu**

(Optional) Specifies to retrieve the peak concurrent usage.

**-cu**

(Optional) Specifies to retrieve current usage.

## himpenv Command-Import

The import utility lets you copy life cycle definitions from one CA Harvest SCM installation to another. This process also copies all access data. If the name of the project exists, himpenv creates a project with the default name.

This command has the following format:

```
himpenv {-b name -f filename} {-usr username -pw password} [-iug] [-v] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-f *filename***

(Required) Specifies a name for the file that will receive the project files.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies Defines the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-iug**

(Optional) Specifies to import a user group for the approve, notify, and linked notify processes to the newly created project.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**Example: himpenv command**

```
himpenv -b broker -usr username -pw password -f filename
```

## hlr Command-Load Repository

The hlr command brings files located in operating system directories into the CA Harvest SCM repository. Only new items can be loaded into a repository. If the items already exist in the repository, an error is generated. Files that are loaded are not associated with any package. They are displayed with the label BASE in the package field.

This command has the following format:

```
hlr {-b name -cp path -rp path -f "names" | pattern} {-usr username -pw password} [-r] [-cep] [-c text] [-rm computername] [-rport portnumber] [-rusr username] [-rpw password] [-v] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename] [-er filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-cp *path***

(Required) Specifies the client path that corresponds to current view path. The client path must be an absolute path.

**-rp *path***

(Required) Specifies the name of the repository in which to load the files and create paths. When you initially load an empty repository, no paths exist. All files to be included in this repository must be loaded beneath this top-level name.



**-f "name" | pattern**

(Required) Specifies a name or a wildcard pattern (\*) to include all matching files in the load operation. Typically, a single wildcard is used, but you can use any number of wildcards in any position. You can also use the question mark (?) for a single character match. The name must be enclosed in quotation marks.

**-usr username**

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw password**

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-r**

(Optional) Specifies the Recursive option, used to load an entire directory structure into a CA Harvest SCM repository. Each directory containing files is created and the files matching the file name pattern loaded. If corresponding paths do not currently exist in the repository, they are created as long as they are not empty.

**-cep**

(Optional) Specifies that the load operation should create matching empty directories in the repository, if any exist, and those containing files. This option is only available when Recursive is selected.

**-c text**

(Optional) Specifies a comment.

**-rm computername**

(Optional) Specifies the remote computer name from which files will be loaded into the repository. Necessary only if you want to load a repository from a remote computer.

**-rport portnumber**

(Optional) Specifies the port number of the remote computer from which files will be loaded into the repository. Necessary only if you want to load a repository from a remote computer.

**-rusr username**

(Optional) Specifies the user name for the remote computer from which files will be loaded into the repository. This option is necessary only if you want to load a repository from a remote computer.

**-rpw *password***

(Optional) Specifies the password for the remote computer from which files will be loaded into the repository. This option is necessary only if you want to load a repository from a remote computer.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-er *filename***

(Optional) Specifies the encrypted remote username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility. -er overrides -musr and -mpw options.

## hlv Command-List Version

The hlv command executes the list version process specified on the command line, or the first list version process in the current context if no process name is specified. The process properties determine the default behavior for the process access and execution of linked processes. Other aspects of the process can be specified through command parameters.

This command has the following format:

```
hlv filename {-b name -en name -st name -vp path} {-usr username -pw password} [-v]
[-vn version] [-pn name] [-s filename | pattern] [-cd | -ac] [-prompt] [-ss name] [-i
inputfile.txt | -di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg]
[-wts] [-h]
```

***filename***

Specifies the file name of the file on which you want to execute the command.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *name***

(Required) Specifies the name of the state in which the user is operating.

**-vp *path***

(Required) Specifies the CA Harvest SCM view path that corresponds to the current working directory.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-vn *version***

(Optional) Specifies a version of an item.

**-pn *name***

(Optional) Specifies a process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-s *filename* | *pattern***

(Optional) Specifies that the list version process should include directories below the current one for files that match a file name or naming pattern.

**-cd**

(Optional) Specifies to display the change description that was provided during check-in.

**-ac**

(Optional) Specifies to display the actual line-by-line changes between one version and the next.

**Note:** The -cd and -ac options default to enabled. If you supply one of the options, the other option is disabled.

**-ss *name***

(Optional, unless the state is a snapshot.) Specifies a snapshot view name. This option is required when the state is a snapshot.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hmvitm Command-Move Item

The hmvitm command executes the move item process specified on the command line, or the first move item process in the current context if no process name is specified. The process definition determines the default behavior and the process access. Other aspects of the process can be specified through command parameters.

This command has the following format:

```
hmvitm item_names {-b name -en name -st name -vp path -np path} {-p name} {usr username
-pw password} [-pn name] [-ot | -ob] [-ur] [-uk] [-de description] [-prompt] [ i
 | -di ] [-eh filename] [-o filename | -oa filename] [-arg]
[-wts] [-h]
```

### ***item\_names***

Specifies the names of the items to be moved.

### **-b *name***

(Required) Specifies the CA Harvest SCM broker name.

### **-en *name***

(Optional) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

### **-st *name***

(Required) Specifies the name of the state in which the user is operating.

### **-vp *path***

(Required) Specifies the CA Harvest SCM view path that contains the item to be moved.

### **-np *path***

(Required) Specifies the name of the path that is the target of the move.

### **-p *name***

(Required) Specifies the CA Harvest SCM package name, read from the command line or a signature file.

### **-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at run time.

### **-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the prompt option.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-ot**

(Optional) Specifies that the new move item version is to be created on the trunk. This parameter is mutually exclusive with **-ob**.

**Note:** This option is only available if the selected or default move item process allows this operation to be applied on the trunk.

**-ob**

(Optional) Specifies that the new move item version is to be created on a branch. This parameter is mutually exclusive with **-ot**.

**Note:** This option is only available if the selected or default move item process allows this operation to be applied on a branch.

**-ur**

(Optional) Creates a normal version for the move item change. If a reserved version already exists in the same package, it is updated as a normal version.

**-uk**

(Optional) Creates a normal version for the move item change. Another new reserved version is created after the normal version.

**-de *description***

(Optional) Specifies a description of the new version created during the move item process. The description is stored in the database.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hmvpkg Command-Move Package

The hmvpkg command executes the first move package process in the second project given. The process properties determine the default behavior for the process access and executing of linked processes. Other aspects of the process can be specified through command parameters read from the command line.

This command has the following format:

```
hmvpkg {-b name -en name -st name} {-usr username -pw password} [-ph] [-pn name] [-ten  
name] [-tst name] [-v] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename]  
[-o filename | -oa filename] [-arg] [-wts] [-h] [-k] package_names...
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *name***

(Required) Specifies the name of the CA Harvest SCM state.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-ph**

(Optional) Specifies to transfer all the history records that have been created in former projects. If the history is no longer required, excluding it can reduce processing time.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-ten *name***

(Optional, unless -tst is specified.) Specifies the CA Harvest SCM destination project. If this argument is specified, the -tst option must also be specified. If this argument is not specified, the CA Harvest SCM administrator must define the target project.

**-tst *name***

(Optional, unless -ten is specified.) Specifies the CA Harvest SCM destination state. If this argument is specified, the -ten option must also be specified. If this argument is not specified, the CA Harvest SCM administrator must define the target state.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-k**

(Optional) Specifies if a duplicate of the source package must be created in the source project for the move.

***package\_names...***

Specifies the names of one or more packages to move.

**Example: hmvpkg command**

```
hmvpkg -usr user -pw password -b broker -en project -st state -ph "Pkg1" "Pkg2" "Pkg3"
```

## hmvproj Command-Move or Copy Projects

The *hmvproj* is a command-line utility for the Harvest SCM Administrator usage only. This utility does not require the Harvest SCM broker connection or client components, to execute. With the specified ODBC connections to both source database and target database, the project is copied or moved between databases.

**Note:** This utility is packaged in the installation of Harvest SCM Server component and not available from client installation.

**Important!** Considering the possible impact to the existing usage, we recommend that you run this utility during off-peak time or your product maintenance period. Depending on the size of project, the copy or move project may take an extended time and database resources to complete the process of query, insert, and update in database. In addition, the verification has been bundled as a post copy project operation to assure the data integrity.

When move project is executed, the source project gets deleted after successful copy and verification of the project. Alternatively, the copy operation can be used and then specify DELETEONLY to remove it from source database.

**Important!** The target database has to be at same or newer Harvest SCM database schema release level than the source database. This utility returns an error, if target database's Harvest SCM release level is older than source database.

This command has the following format:

```
hmvproj { -s_odbc source odbc name } { -s_usr username -s_pw password } { -t_odbc target odbc name } { -t_usr username -t_pw password } [-s_eh source encrypted filename ] [-t_eh target encrypted filename ] { -pj projectName } { -user_opt option } { -form_opt option } [-verifyonly ] [-verifyonly length ] [ -verifyrollback ] [-delete] [-deleteonly] [-h]
```

**-S\_ODBC**

Specifies the ODBC name for the source project database.

**-S\_USR**

Specifies the source database login username.

**-S\_PW**

Specifies the source database login password.

**-T\_ODBC**

Specifies the ODBC name of the target project database.

**-T\_USR**

Specifies the target database login username.

**-T\_PW**

Specifies the target database login password.

**-S\_EH**

Specifies the source database encrypted login username/password.

**-T\_EH**

Specifies the target database encrypted login username/password.

**Note:** The encrypted database credentials for both the source and target databases use the convention that the server install is supported.



**-PJ**

Specifies the SCM project name.

**-USER\_OPT**

Specifies the user/user group handling option flag.

Default valid values: 0, 1, 2

**0**

Copy all users and user groups.

**1**

Copy only users and user groups, which has access to that project.

**2**

Ignore user and usergroup.

**Note:** If you select this value:

- All the relationship between user/usergroup and other objects in this project gets dropped.  
  
For example, all access settings, the approve process user list, and the notify process user list are dropped.
- If the same name of a user or usergroup exists in the target database, their data does not get updated.

**-FORM\_OPT**

Specifies the form handling option flag.

Default valid values: 0, 1, 2

**0**

Ignore form if exists in target database.

**1**

Overwrite form if exists in target database and source database has newer updated date.

**2**

Always overwrite form if exists in target database.

**-DELETE**

Specifies the option to delete the project in the source database after completing the copy operation.

**-DELETEONLY**

Specifies the option to delete the project in the source database directly without copying.

#### **-VERIFYDATA**

Specifies the option to verify the version data and the amount of data that must be compared for each version data record.

For example, specifying `-VERIFYDATA=4096` indicates that the version data must be verified and that for each version data record. The data comparison is done for every 4096 bytes of total data records.

Specifying `-verifydata=0` indicates that the entire version data must be compared for each record.

**Note:** The `-verifydata` option is used in a separate run of `hmvproj` after a project has already been copied successfully.

#### **-VERIFYROLLBACK**

Specifies the option of rolling back the copied project, if verify fails.

#### **-SQL**

Specifies the option of SQL getting copied in to the log file for debugging.

## hmvpth Command-Move Path

The `hmvpth` command executes the move path process specified on the command line, or the first move path process in the current context if no process name is specified. The process definition determines the default behavior and the process access. Other aspects of the process can be specified through command parameters.

This command has the following format:

```
hmvpth {-b name -en name -st name -vp path -np path} {-p name} {-usr username -pw password} [-pn name] [-ot | -ob] [-ur] [-uk] [-de description] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

#### **-b name**

(Required) Specifies the CA Harvest SCM broker name.

#### **-en name**

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

#### **-st name**

(Required) Specifies the name of the state in which the user is operating.

**-vp *path***

(Required) Specifies the CA Harvest SCM view path that is to be moved.

**-np *path***

(Required) Specifies the name of the path that is the target of the move.

**-p *name***

(Required) Specifies the CA Harvest SCM package name, read from the command line or a signature file.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the prompt option.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-ot**

(Optional) Specifies that the new move path version and all of its subitem versions are to be created on the trunk. This parameter is mutually exclusive with -ob.

**Note:** This option is only available if the selected or default move path process allows this operation to be applied on the trunk.

**-ob**

(Optional) Specifies that the new move path version and all of its subitem versions are to be created on a branch. This parameter is mutually exclusive with -ot.

**Note:** This option is only available if the selected or default move path process allows this operation to be applied on a branch.

**-ur**

(Optional) Creates a normal version for the move path change. If a reserved version already exists in the same package, it is updated as a normal version.

**-uk**

(Optional) Creates a normal version for the move path change. Another new reserved version is created after the normal version.

**-de *description***

(Optional) Specifies a description of the new version created during the move path process. The description is stored in the database.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hpg Command-Package Group Manager

The hpg command lets you maintain packages and package groups in CA Harvest SCM package groups. You can create package groups (-cpg), delete package groups (-dpg), add packages to a package group (-app), and delete packages from a package group (-dpp). If you specify a state (-st), only packages in that state are considered. If a package group is deleted, all packages that are associated with the package group will also be deleted from the group. Note that the Create Package Group and Delete Package Group options are mutually exclusive.

This command has the following format:

```
hpg {-b name -en name -pg name} {-usr username -pw password} [-st name] [-bp] [-cpg]
[-dpg] [-app names...] [-dpp names...] [-v] [-prompt] [-i inputfile.txt]
[-di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-pg *name***

(Required) Specifies a CA Harvest SCM package group.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-st *name***

(Optional) Specifies the name of the state in which hpg is operating. If you do not specify a state with this option, the package list is retrieved from the specified environment.

**-bp**

(Optional) Specifies that the bind package restrictions should be enforced with packages belonging to the package group.

**-cpg**

(Optional) Specifies to create a package group using the name specified with the -pg option.

**-dpg**

(Optional) Specifies to delete the package group named with the -pg option.

**-app *names...***

(Optional) Specifies to add one or more packages to the package group named with the -pg option. You can specify exact names, or use the wildcard (\*) or placeholder (?) characters.

**-dpp *names...***

(Optional) Specifies to delete packages from the package group named with the -pg option. You can specify exact names, or use the wildcard (\*) or placeholder (?) characters.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hpkgunlk Command-Unlock Package

The hpkgunlk command lets administrators unlock packages that have been locked in a non-idle status. If a package retains its in-use status and gets locked (preventing users from performing any further processes on the package), an administrator must first verify that the package is not in use by another CA Harvest SCM user. The administrator can then use the hpkgunlk command to reset the status of a locked package and make it available for processes.

**Note:** This command is for use by administrators only. Use this command with caution and only after verifying the package in question is in a non-idle status and no other users are using the package.

This command has the following format:

```
hpkgunlk package_name {-b name -en name} {-usr username -pw password} [-v] [-prompt]
[-i inputfile.txt | -di inputfile.txt] [-eh filename] [-o filename or -oa filename]
[-wts] [-h]
```

*package\_name*

Specifies the name of the package to be unlocked.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-usr *username***

(Required, unless -prompt specified.) Specifies user credentials.

**Note:** You can also specify user credentials using the -eh option or with the user name (-usr) and password (-pw) options.

**-pw *password***

(Required, unless -prompt specified.) See -usr and -eh descriptions.

**-v**

(Optional) Displays the relevant variables used for the command execution.

**-prompt**

(Optional) Prompts you for your -usr and -pw.

**-o *filename***

(Optional) Specifies a file name to receive the output log. If a file of the same name exists, it is replaced.

**-oa *filename***

(Optional) Specifies a file name to receive the output log. If a file of the same name exists, appends the log information.

**-wts**

(Optional) Records the time of command line execution in the output log.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username/password file name. The file name used with this option contains encrypted username/password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hpp Command-Promote Package

The hpp command executes the promote process specified on the command line, or the first promote process in the current context if no process name is specified. The process definition determines the default behavior, process access, and execution of linked processes. Other aspects of the process can be specified through command parameters.

This command has the following format:

```
hpp {package_names... | -pg packagegroup_names...} {-b name -en name -st name} {-usr  
username -pw password} [-pb] [-pm] [-pd] [-v] [-pn name] [-prompt] [-i inputfile.txt  
| -di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-pdr]  
[-vdr] [-adp] [-h]
```

***package\_names...* | -pg *packagegroup\_names...***

Specifies one or more names of packages to be promoted. Or, with the -pg option, specifies one or more package group names to be promoted.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *name***

(Required) Specifies the name of the CA Harvest SCM state in which the user is operating.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-pb**

(Optional) Specifies that all the packages belonging to a bound package group must be promoted together.

**-pm**

(Optional) Specifying this option prohibits packages to be promoted to the next state if they are merge-tagged, reserved-tagged, or associated with branch versions. If you enforce that packages must be merged, two requirements must be met:

- The latest change for the package must be on the trunk, not on a branch.
- The latest package version must not be tagged as merged (M) or reserved (R). The interactive merge process must be used to resolve the merge tag before the package can be promoted.

**-pd**

(Optional) Specifying this option disallows promotion of packages that depend upon other packages. Verifies package dependency. Dependency is based on versions in the view. In the current state, a package with a higher item-version cannot be promoted without also promoting the packages with the lower item-versions in the current view, unless the lower item-versions already exist in the view of the Promote To state. If the promote process is in a state that shares the same view as the one in the Promote To state, then this option is not enforced.

**Note:** The lower item-version will only cause a dependency error when it is on the trunk.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.



**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-pdr**

(Optional) Generates the package dependency report by listing all dependent packages. The Promote process is not executed when this option is specified.

**-vdr**

(Optional) Generates the package dependency report by listing all dependent versions. The Promote process is not executed when this option is specified.

**-adp**

(Optional) Adds dependent packages in the current state to the execution list, when specified.

## hpolget Command-Get Password Policy

The hpolget command generates a password policy configuration file. If the invoking user is in the Administrator user group, all user-level overrides are included; otherwise, only the overrides that apply to the invoking user are reported. The following considerations apply to authentication:

- When you are using internal authentication (CA Harvest SCM authentication), retrieving the password policy is enabled.
- When you are using external authentication, such as Microsoft Active Directory, this functionality is disabled; retrieving password policy settings must be performed using methods provided by the authentication server. For example, if using Microsoft Active Directory, you can use MMC + Group Policy Snap-in to retrieve password policy settings.
- When you are using external authentication, password policy retrieval fails with an error message.

We recommend that you record the name of the user who last modified password policy and the time modified in a header or comment in the configuration file.

After you generate the password policy configuration file, use the hppolset command to set it.

This command has the following format:

```
hppolget {-b name} [-usr username] [-pw password] [-f policy_config_file] [-v]  
[-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename] [-o filename | -oa  
filename] [-arg] [-gl] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Optional) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Optional) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-f *policy\_config\_file***

(Required) Specifies the name of a policy configuration file to which parsed keys are exported for archiving or review.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-gl**

(Optional) Specifies that the time format will be output according to the local settings.

## hppolset Command-Set Password Policy

The hppolset command sets the password policy configuration file (*policy\_config\_file* in the following syntax). Password policy rules are read from the configuration file.

The following considerations apply to authentication:

- When you are using internal authentication (CA Harvest SCM authentication), updating password policy is enabled.
- When you are using external authentication, such as Microsoft Active Directory, this functionality is disabled; updating password policy settings must be performed using methods provided by the authentication server. For example, if using Microsoft Active Directory, you can use MMC + Group Policy Snap-in to update security policy settings.
- When you are using external authentication, password policy updating fails with an error message.

This command has the following format:

```
hppolset {-b name} [-usr username] [-pw password] [-fc] [-v] [-prompt] [-i inputfile.txt |  
-di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-f policy_config_file] [-arg]  
[-wts] [-h]
```

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-usr username**

(Optional) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw password**

(Optional) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-fc**

(Optional) Specifies Force Change, which applies the “Change on next logon” override to all users for whom the “Never expire” override has not been applied. This forces users to comply with new password policy. This setting overrides the contents of the *policy\_config\_file* specified in -f.

**-eh filename**

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-f *policy\_config\_file***

(Required) Specifies the name of a policy configuration file. All password policy settings are read from this file.

## Password Policy

Password policy is a set of user-defined rules governing password usage. This feature enables CA Harvest SCM administrators to configure and validate passwords according to the security requirements of the installation environment. To facilitate this, the password policy includes the most common types of password rules.

CA Harvest SCM password policy consists of two types of password rules.

- **Global rules** apply to all CA Harvest SCM users including users in the Administrator user group.
- **User-level override rules** apply to specific users and override one or more global rules.

## Policy Configuration File

A policy configuration file contains the keys and values for specifying the password policy. Each line in the file represents a configuration option and its value. Configuration options can be global, applying to all users, or user-level, applying to an individual user.

## Global Options

Global options are configuration values that apply to all users. Global options can be overridden for individual users by specifying user-level overrides.

Each global option has the following format:

*option* = value

The following rules apply to global options:

- Global option names are case insensitive.
- User names are case sensitive.
- Empty lines are allowed.
- Invalid option names will return an error.
- Comments must begin with a “#” character.
- If multiple lines defining the same option exist, the last line takes precedence.
- Boolean values may be TRUE, FALSE, T, or F.

**MaximumPasswordAge**

Specifies the maximum age in days for a password. If the password is older than this value, the user must change his or her password. To disable maximum password age checking, set the value to 0.

**Example: MaximumPasswordAge global option**

```
maximumpasswordage = 9
```

**MinimumPasswordAge**

Specifies the minimum age in days for a password. User must have this many days between attempts to change the password. To disable minimum password age checking, set the value to 0.

**Example: MinimumPasswordAge global option**

```
minimumpasswordage = 2
```

**MinimumPasswordLength**

Specifies the minimum character length of passwords. To disable minimum password length checking, set the value to 0.

**Example: MinimumPasswordLength global option**

```
minimumpasswordlength = 8
```

**PasswordCountBeforeReusable**

Specifies the number of unique passwords that must be used before a password may be repeated. To disable password counting, set the value to 0.

**Limit:** Maximum count allowed is 24.

**Example: PasswordCountBeforeReusable global option**

```
passwordcountbeforereusable = 10
```

**MaxFailAttemptBeforeLockout**

Specifies the number of invalid password attempts before the account is locked out. Administrators must unlock the account after that. To disable Maximum Failing Attempts checking, set the value to 0.

**Example: MaxFailAttemptBeforeLockout global option**

```
maxfailattemptbeforelockout = 3
```

**AllowChangeAfterExpire**

Specifies a boolean value to determine whether users can change a password after it has expired. If not, then after it has expired an Administrator must unlock the account. To always allow changes, set the value to true.

**Example: AllowChangeAfterExpire global option**

```
allowchangeafterexpire = true
```

### **MaximumRepeatableCharacter**

Maximum length of string in password made up of a single repeated character. To allow any length of repeating characters, set to 0.

#### **Example: MaximumRepeatableCharacter global option**

```
maximumrepeatablecharacter = 3
```

This setting allows a password such as “paswdaaa” with a string of 3 a's.

### **ExpirationWarningAge**

Specifies the password age in days at which CA Harvest SCM will start issuing warnings that the password will expire. To have no warning, set to 0.

#### **Example: ExpirationWarningAge global option**

```
expirationwarningage = 3
```

### **MinimumNumericCharacter**

Specifies the minimum number of numeric characters that must be in the password.

#### **Example: MinimumNumericCharacter global option**

```
minimumnumericcharacter = 1
```

### **MinimumLowercaseCharacter**

Specifies the minimum number of lowercase characters that must be in the password.

#### **Example: MinimumLowercaseCharacter global option**

```
minimumlowercasecharacter = 1
```

### **MinimumUppercaseCharacter**

Specifies the minimum number of uppercase characters that must be in the password.

#### **Example: MinimumUppercaseCharacter global option**

```
minimumuppercasecharacter = 2
```

### **MinimumNonalphanumericCharacter**

Specifies the minimum number of nonalphanumeric characters that must be in the password.

#### **Example: MinimumNonalphanumericCharacter global option**

```
minimumnonalphanumericcharacter = 0
```

**AllowUsernameAsPassword**

Specifies a boolean value to determine if the user's password may match his or her own username. To always allow the user name to match the password, set the value to true.

**Example: AllowUsernameAsPassword global option**

```
allowusernameaspassword = false
```

**ForceChange**

Specifies that *all* users be forced to change their password on next login, to ensure that all passwords conform to the active password policy. Password policy is not applied retroactively to existing passwords.

**Example: ForceChange Global Option**

```
forcechange= true
```

**Example: Password policy configuration file**

```
# CA Harvest SCM Password Policy Configuration 03/08/2007;16:15:44
```

```
# Global Options
maximumpasswordage = 9
minimumpasswordage = 2
minimumpasswordlength = 8
passwordcountbeforereusable = 10
maxfailattemptbeforelockout = 3
maximumrepeatablecharacter = 2
expirationwarningage = 3
minimumnumericcharacter = 1
minimumlowercasecharacter = 1
minimumuppercasecharacter = 2
minimumnonalphanumericcharacter = 1
allowusernameaspassword = false
forcechange=true
```

```
# User-Level Overrides
paul.passwordneverexpire = true
scott.passwordchangenextlogon = true
```

## User-Level Overrides

Preceding a global option by a user name designates a user-level override. A user-level override is a password rule that is applied to a specific user and overrides one or more global password rules.

Each user-level override has the following format:

```
user.option = value
```

The following user-level overrides are mutually exclusive. For a specific user, only one override setting can be true.

#### **PasswordNeverExpire**

Specifies that a user is to be exempt from the MaximumPasswordAge rule. If this rule is set to true, the user's password will never expire.

##### **Example: PasswordNeverExpire user-level override**

```
username.passwordneverexpire = true
```

#### **PasswordChangeNextLogon**

Specifies that a user's password is set to immediately expire. If this rule is set to true, the user is required to change password at next login. The MinimumPasswordAge and AllowChangeAfterExpire rules are not applicable to the user. This user-level override is removed after the user successfully changes the password.

##### **Example: PasswordChangeNextLogon user-level override**

```
username.passwordchangenextlogon = true
```

**Note:** To disable a user-level override, it must be set to false. Deleting a user-level override statement line from the configuration file and running hppolset does **not** disable the rule.

## **Access to the Policy Configuration File**

Only users in the Administrator user group can set global password rules and unlock locked out users. However, any user who can log in to CA Harvest SCM can view the password global rules. Any user granted with Admin User access method can set user-level overrides for users who are not in the Administrator user group. User-level overrides can also be viewed by anyone with View User access method.

## **Usage**

Password policy rules are optional. When CA Harvest SCM is first installed, two rules are enabled: MinimumPasswordLength is initially set to 6, and AllowChangeAfterExpire is set to yes. A new CA Harvest SCM installation has only one user: harvest. The harvest password should be changed before changing the settings for MaximumPasswordAge and AllowChangeAfterExpire. Otherwise, the harvest user will be locked out. If harvest is the only user, no one is able to unlock harvest.



The following rules are enforced when a new password is created:

- MinimumPasswordLength
- MaximumRepeatableCharacter
- MinimumNumericCharacter
- MinimumLowercaseCharacter
- MinimumUppercaseCharacter
- MinimumNonalphanumericCharacter
- AllowUsernameAsPassword
- MinimumPasswordAge
- PasswordCountBeforeReusable

The other rules are enacted at login time.

Some rules are intended for use in conjunction with other rules:

- The MaximumPasswordAge rule is not useful unless the PasswordCountBeforeReusable rule has been set. The PasswordCountBeforeReusable rule is intended to prevent users from bypassing the intended effect of the MaximumPasswordAge. For example, if MaximumPasswordAge is set to 5 and PasswordCountBeforeReusable is set to 1, 10 days must pass before the user can reuse a password (assuming the user changes his password only when it is expired by CA Harvest SCM).
- The PasswordCountBeforeReusable rule is not useful unless the MinimumPasswordAge rule has been set. The PasswordCountBeforeReusable rule helps prevent users from bypassing the intended effect of the MinimumPasswordAge. For example, if PasswordCountBeforeReusable is set to 6 and MinimumPasswordAge is set to 1, a user must wait 6 days in order to reuse her original password (assuming the user changes her password each day).
- Rules governing the treatment of expired passwords cannot be enacted unless MaximumPasswordAge is specified. If MaximumPasswordAge is not specified, passwords do not expire.

**More information:**

[husrunk Command-Unlock User](#) (see page 170)

## Example: Policy Configuration File

In the following example, users are governed by the following password policy:

- Can change their expired password (AllowChangeAfterExpire = true)
- Cannot use their user name (AllowUsernameAsPassword = false)
- Warnings are issued after 30 days (ExpirationWarningAge = 30)
- Locked out after three failed logons (MaxFailAttemptBeforeLockout = 3)
- Passwords expire after 45 days (MaximumPasswordAge = 45)
- Allowed to repeat characters (MaximumRepeatableCharacter = 0)
- Do not need a lowercase character (MinimumLowercaseCharacter = 0)
- Do not need a special character (MinimumNonalphanumericCharacter = 0)
- Do not need a numeric character (MinimumNumericCharacter = 0)
- Five days before changing passwords (MinimumPasswordAge = 5)
- Passwords must be six characters or more (MinimumPasswordLength = 6)
- Do not need an uppercase character (MinimumUppercaseCharacter = 0)
- Cannot reuse the previous three passwords (PasswordCountBeforeReusable = 3)

To set this policy, specify the following rules in the configuration file:

```
AllowChangeAfterExpire = true
AllowUsernameAsPassword = false
ExpirationWarningAge = 30
MaxFailAttemptBeforeLockout = 3
MaximumPasswordAge = 45
MaximumRepeatableCharacter = 0
MinimumLowercaseCharacter = 0
MinimumNonalphanumericCharacter = 0
MinimumNumericCharacter = 0
MinimumPasswordAge = 5
MinimumPasswordLength = 6
MinimumUppercaseCharacter = 0
PasswordCountBeforeReusable = 3
```

User-level overrides can be applied to individual users and are added to the password configuration file. Two user-level overrides are available: PasswordChangeNextLogon and PasswordNeverExpire. Pre-pending the user name of the individual user in front of the rule sets these password policy rules. In the following example:

- User paul must change his password at his next login.
- User maria's password will never expire.

The password policy configuration file user-level override appears in the following manner:

```
# User-level Overrides

paul.PasswordChangeNextLogon = true
maria.PasswordNeverExpire = true
```

## hrefresh Command-Refresh Directories

The hrefresh command works with hsync to populate reference directories with appropriate versions of files from CA Harvest SCM views. The hrefresh command is installed with the CA Harvest SCM server and it is designed to be executed in a CA Harvest SCM server UDP.

**Note:** Concepts that are key to understanding the hrefresh command are described in the *Administrator Guide*. The *Administrator Guide* also describes how to set up and configure hrefresh, and includes detailed information about the -nolock option.

The hrefresh command reads standard input (STDIN) where you can specify a list of packages or a list of versions. The hrefresh command reads a configuration file to determine what reference directories to update, and how to update them. The utility writes output to log files in the hrefresh home directory.

**Note:** The return codes described in the appendix "Return Codes" do not apply to hrefresh. The hrefresh command exits with return code 1 if there is a fatal error. Otherwise, it returns 0.

This command has the following format:

```
hrefresh -b broker -pr project -st state [-nst nextstate] [-iv | -pl | -ps | -pv] [-h]
[-debug] [-nolock] [-cfg configfilename] -execmode [synchronous|asynchronous]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-pr *name***

(Required) Specifies the CA Harvest SCM project name.

**-st *name***

(Required) Specifies the name of the state in which the user is operating.

**-nst *nextstate***

(Optional) Specifies the name of the next state. This option is intended for use with the [nextstate] system variable. This variable has a value in UDPs that are post-linked to Promote or Demote states. If you specify *nextstate*, hrefresh will execute hsync in that state and the current state. If you do not provide *nextstate*, hrefresh ignores the -nst option.

**-iv**

(Optional) Sets the item selection option to All Items in View, which selects all items under a given view path. You can use this option in a stand-alone UDP. It is the default setting for item selection. The -iv item selection option overrides data from standard input.

**Note:** Item selection options are described in the hsync command topics.

**-pl**

(Optional) Sets item selection option to Package List. Standard input may contain a list of packages, a list of versions, or both. By default, when standard input contains both packages and versions, hrefresh will use the version list and ignore the package list. This option explicitly specifies that a package list is to be used, ignoring the version list.

**Note:** The item selection options, -iv, -pl, -ps, and -pv, are mutually exclusive.

**-ps**

(Optional) Sets the item selection option to All Packages in State. This option selects items in packages that are located in the state where hsync is executing. Use this option as an efficient alternative to the package list when the set of packages is large and all the packages are in the given state. The -ps item selection option overrides data from standard input.

**Note:** The item selection options, -iv, -pl, -ps, and -pv, are mutually exclusive.

**-pv**

(Optional) Sets the item selection option to All Packages in View. This option selects items in packages that are located in any state that shares a view with the packages' current state. This is an alternative to use when the packages are in states that share a view. The -pv item selection option overrides data from standard input.

**Note:** The item selection options, -iv, -pl, -ps, and -pv, are mutually exclusive.

**-debug**

(Optional) Runs the program in debug mode and prints all debug values to the standard output.

**-nolock**

(Optional) Overrides semaphore locks, thereby allowing multiple check-out operations for the same target directory to run concurrently. If you do not specify -nolock, the semaphore locks are active (the default), and these operations run consecutively.

**Note:** When -nolock is in effect, the -maxretries and -retrydelay settings in the HRefresh.arg file are ignored.

Use -nolock with caution. For details, see the *Administrator Guide*.

To override semaphore locks, specify the -nolock option as follows:

- To set -nolock for certain instances of hrefresh, specify -nolock when you run hrefresh from the command line, as explained in this section.
- To set -nolock for all hrefresh executions, add a line specifying -nolock to your HRefresh.arg file. For details, see the *Administrator Guide*.

**-cfg filename**

(Optional) Specifies the configuration file name. For more information about creating a hrefresh configuration file, see the *Administrator Guide*.

**Default:** hrefresh.cfg

**execmode synchronous|asynchronous**

(Optional) Sets the default execution mode for all hrefresh executions, unless overridden by using -execmode on the hrefresh command line.

**Default:** asynchronous

**More information:**

[hsync Command-Synchronize Directory from View Path](#) (see page 151)

## hredit Command-Edit Repository

The hredit command allows administrators to:

- Move items to another location in a repository
- Rename repositories
- Create empty repository paths
- Rename paths

This command has the following format:

```
hrepedit {-b name -rp name} {-ppath -tpath} {{-rnpath} -oldname old_rep_name -newname  
new_rep_name } {-usr username -pw password} [-all] [-fo] [-ismv] [-isren] [-v]  
[-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename] [-o filename | -oa  
filename] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-rp *name***

(Required) Specifies the empty path name that will be created in the repository, or the parent path when renaming item paths.

**-ppath**

(Required for move item.) An option for move repository item. Use this option to specify the parent path of the items being moved.

**-tpath**

(Required for move item.) An option for move repository item. Use this option to specify the target item path.

**-rnpath**

(Required for rename path.) Specifies to rename an item path from *old\_path\_name* to *new\_path\_name*.

**-oldname *old\_rep\_name***

(Required for rename repository or rename path.) Specifies the name of the repository or item path that you want to rename.

**-newname *new\_rep\_name***

(Required for rename repository or rename path.) Specifies the new name of the repository or item path that you are renaming.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-all**

(Optional) Specifies all baseline, an option for create empty path. This option creates the path structure in all existing baselines of the repository.

**Note:** -all and -fo cannot be specified at the same time. If neither is specified, -fo is the default option.

**-fo**

(Optional) Specifies future only baseline, an option for create empty path. This option only creates the path structure in the repository and affects only future baselines.

**Note:** -all and -fo cannot be specified at the same time. If neither is specified, -fo is the default option.

**-ismv**

(Optional for move repository item and rename path.) Specifies that the CA Harvest SCM server should move the repository items without checking the snapshot view. If -ismv is not specified, the CA Harvest SCM server checks if any item being moved is associated with any snapshot view. If so, the CA Harvest SCM server only returns a warning message without moving items.

**-isren**

(Optional for rename repository.) Specifies that the CA Harvest SCM server should rename the repository without checking the snapshot view. If -isren is not specified, the CA Harvest SCM server checks if any snapshot view is associated with the repository being renamed. If so, the CA Harvest SCM server only returns a warning message without renaming the repository.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**Example: hrepedit command**

The following are examples of the hrepedit command with various options:

**Rename repository**

To rename the repository from *old\_rep\_name* to *new\_rep\_name*:

```
hrepedit -b brokername -usr username -pw password -oldname old_rep_name -newname  
new_rep_name -isren
```

### Rename item path

To rename an item path from *old\_path\_name* to *new\_path\_name*:

```
hrepedit -b brokername -usr username -pw password -rp repository -rnp oldname  
old_path_name -newname new_path_name -isren
```

### Move item

To move item1, item2, and item3 from \TestRep\dir1 to \TestRep\dir2:

```
hrepedit -b brokername -usr username -pw password -ppath  
\TestRep\dir1 -tpath \TestRep\dir2 -ismv "item1" "item2" "item3"
```

### Create empty path

To create an empty path dir3 under \TestRep with the future only option:

```
hrepedit -b brokername -usr username -pw password -rp \TestRep\dir3 -fo
```

### Move directory

To move dir1 from \TestRep\ to \TestRep\dir2:

```
hrepedit -b brokername -usr username -pw password -ppath  
\TestRep\ -tpath \TestRep\dir2 -ismv "dir1"
```

## hrepmgr Command-Manage Repository

The hrepmgr command is used for repository management. The hrepmgr command lets administrators do the following functions:

- Create repositories
- Delete repositories
- Duplicate repositories
- Load repositories
- Rename repositories
- Update repository property
- Delete item or path
- Move item or path
- Rename item or path
- Update item or path property
- Create empty path

These operations are mutually exclusive.

**Note:** When a pattern is used in the command, it must be in single or double quotation marks in UNIX.



## Create Repository

The option `-cr` is used with `hrepmgr` to create a new repository, and has the following syntax:

```
hrepmgr name(s) -cr{options}
```

**-cr**

(Required) Specifies to create repository or path.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Required, unless `-prompt` is specified) Specifies the user name for login to the broker. If `-usr` is defined without `-pw`, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless `-prompt` is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the `-prompt` option.

**-nc**

(Optional) Specifies that binary files should not be compressed.

**-coe *ext(s)***

(Optional) Specifies that all binary files should be compressed except for those with the extensions specified.

**-mvs**

(Optional) Specifies that MVS is set to *on*.

**-noext**

(Optional) Specifies that file extensions should not be used.

**-rext**

(Optional) Specifies that the repository text file extension should be used.

**-addext *ext(s)***

(Optional) Specifies to add an extension. It must be used with `-rext` option.

**-addvgrp *group(s)***

(Optional) Adds view group. Options for adding and removing group(s) for view access can be used together.

**-addugrp *group(s)***

(Optional) Adds update group. Options for adding and removing group(s) for update access can be used together.

**-addsgrp *group(s)***

(Optional) Adds secure group. Options for adding and removing group(s) for secure access can be used together.

**-c "*comment*"**

(Optional) Specifies to add a comment to the repository notes field.

**Note:** The options -c and -fc are mutually exclusive; you can use one or the other.

**-fc *path\_file.txt***

(Optional) Specifies to add comments from a file to the repository notes field.

**Note:** The options -c and -fc are mutually exclusive; you can use one or the other.

**Example: Create Repository**

To create a new repository TestRep with default property:

```
hrepmgr -b brokername -usr username -pw password -cr TestRep
```

## Delete Repository

The option -del is used with hrepmgr to delete a repository, and has the following syntax:

```
hrepmgr name(s) or "pattern" -del{options}
```

**-del**

(Required) Specifies to delete a repository.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**Example: Delete Repository**

To delete a repository TestRep:

```
hrepmgr -b brokername -usr username -pw password -del TestRep
```

## Duplicate Repository

The option -dup is used with hrepmgr to duplicate a repository, and has the following syntax:

```
hrepmgr -dup -srn name -drn name {options}
```

**-dup**

(Required) Specifies to duplicate a repository. The options -srn and -drn must be used with this option.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-srn *name***

(Required) Specifies the source repository name for the -dup (duplicate repository) option.

**-drn *name***

(Required) Specifies the destination repository for the -dup (duplicate repository) option.

**-ndac**

(Optional) Specifies that access control should not be duplicated for the -dup (duplicate repository) option.

### Example: Duplicate Repository

To duplicate repository TestRep2 from repository TestRep1:

```
hrepmgr -b brokername -usr username -pw password -dup -srn TestRep1 -drn TestRep2
```

## Load Repository

Use the `-ld` option with `hrepmgr` to load files from a repository. Use the following syntax:

```
hrepmgr "pattern" -ld -cp clientpath -rp repositorypath {options}
```

### **-ld**

(Required) Specifies to load a repository.

**Note:** The `-cp` option must be used with this option.

### **-b *name***

(Required) Specifies the CA Harvest SCM broker name.

### **-usr *username***

(Required, unless `-prompt` is specified) Specifies the user name for login to the broker. If `-usr` is defined without `-pw`, the user is prompted for the password at runtime.

### **-pw *password***

(Required, unless `-prompt` is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the `-prompt` option.

### **-cp *clientpath***

(Required) Specifies a client path for the `-ld` (load repository) option.

### **-rp *repositorypath***

(Required) Specifies the repository path.

### **-r**

(Optional) Specifies to recursively load files.

### **-cep**

(Optional) Specifies to create empty repository paths during recursive loading.

**-rm**

(Optional) Specifies the remote computer name.

**-rport**

(Optional) Specifies the remote computer port number.

**-rusr**

(Optional) Specifies the remote user.

**-rpw**

(Optional) Specifies the remote user password.

**-er *filename***

(Optional) Specifies the remote user filename.

**-c "*comment*"**

(Optional) Specifies to add a comment to the repository notes field.

**Note:** The options -c and -fc are mutually exclusive; you can use one or the other.

**Example: Load Repository**

To recursively load all files from the client path \TestRep1\dir1 to repository path \TestRep2\dir2:

```
hrepmgr -b brokername -usr username -pw password -ld "*" -cp "\TestRep1\dir1" -rp \TestRep2\dir2 -r
```

## Rename Repository

The option -ren, is used with hrepmgr to rename a repository, and has the following syntax:

```
hrepmgr -ren -oldname name -newname name {options}
```

**-ren**

(Required) Specifies to rename a repository. The options -rp, -oldname, and -newname can be used with this option.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-oldname *name***

(Required) Specifies the name of the repository to rename.

**-newname *name***

(Required) Specifies the new name of the repository being renamed.

**-isv**

(Optional) Specifies to ignore the snapshot view for the rename repository.

**Example: Rename Repository**

To rename the repository TestRep1 to TestRep2:

```
hrepmgr -b brokername -usr username -pw password -ren -oldname TestRep1 -newname  
TestRep2
```

## Update Repository Property

The option -upd, is used with hrepmgr to update a repository property, and has the following syntax:

```
hrepmgr name(s) or "pattern" -upd {options}
```

**-upd**

(Required) Specifies to update a repository.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-nc**

(Optional) Specifies that binary files should not be compressed.

**-coe *ext(s)***

(Optional) Specifies that all binary files should be compressed except for those with the extensions specified.

**-co**

(Optional) Compresses all binary files.

**-appext**

(Optional) Specifies to append an extension or extensions to the extension list.

**-mvs**

(Optional) Specifies that MVS is set to *on*.

**-nmvs**

(Optional) Specifies that MVS is set to *off*.

**-gext**

(Optional) Specifies to use global repository text file extension.

**-noext**

(Optional) Specifies that file extensions should not be used.

**-rext**

(Optional) Specifies that the repository text file extension should be used.

**-addext *ext(s)***

(Optional) Specifies to add an extension.

**Note:** This option must be used with the -rext option.

**-remext *ext(s)***

(Optional) Specifies to remove an extension or extensions.

**-advgrp *groupname(s)***

(Optional) Specifies to add view group.

**Note:** Options for adding and removing group(s) for view access can be used together.

**-remvgrp *groupname(s)***

(Optional) Specifies to remove view group.

**Note:** Options for adding and removing group(s) for view access can be used together.

**-addugrp *groupname(s)***

(Optional) Specifies to add update group.

**Note:** Options for adding and removing group(s) for update access can be used together.

**-remugrp *groupname(s)***

(Optional) Specifies to remove update group.

**Note:** Options for adding and removing group(s) for update access can be used together.

**-addsgrp *groupname(s)***

(Optional) Specifies to add one or more secure groups.

**Note:** Options for adding and removing group(s) for secure access can be used together.

**-remsgrp *groupname(s)***

(Optional) Specifies to remove one or more secure groups.

**Note:** Options for adding and removing group(s) for secure access can be used together.

**-c "*comment*"**

(Optional) Specifies to add a comment to the repository notes field.

**Note:** The options -c and -fc are mutually exclusive; you can use one or the other.

**-fc *path\_file.txt***

(Optional) Specifies to add comments from a file to the repository notes field.

**Note:** The options -c and -fc are mutually exclusive; you can use one or the other.

**-appc**

(Optional) Specifies to append comments to repository note field.

**Example: Update Repository Property**

To set the property to compress all binary files in TestRep1:

```
hrepmgr -b brokername -usr username -pw password -upd TestRep1 -co
```



## Move Item or Path

The option `-mv`, is used with `hrepmgr` to move an item or path and has the following syntax:

```
hrepmgr itemname(s) or pathname(s) or "pattern" -mv -srp path  
-drp path {options}
```

**-mv *itemname* | *pathname* | "*pattern*"**

(Required) Specifies to move an item or path.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Required, unless `-prompt` is specified) Specifies the user name for login to the broker. If `-usr` is defined without `-pw`, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless `-prompt` is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the `-prompt` option.

**-srp *path***

(Required) Specifies the source repository path name for the `-mv` (move item or path) option.

**-drp *path***

(Required) Specifies the destination repository path name for the `-mv` (move item or path) option.

**-isv**

(Optional) Specifies to ignore the snapshot view for the move item or path.

### Example: Move Item or Path

To move the item `TestItem` and the path `TestPath` from repository path `\Rep\Path1` to the repository path `\Rep\Path2` without checking the snapshot view:

```
hrepmgr -b brokername -usr username -pw password -mv TestItem TestPath -srp  
\Rep\Path1 -drp \Rep\Path2 -isv
```

## Delete Item or Path

The option `-del`, is used with `hrepmgr` to delete an item or repository path and has the following syntax:

```
hrepmgr itemname(s) or pathname(s) or "pattern" -del -rp path {options}
```

### **-del**

(Required) Specifies to delete a repository item, or path.

### **-b *name***

(Required) Specifies the CA Harvest SCM broker name.

### **-usr *username***

(Required, unless `-prompt` is specified) Specifies the user name for login to the broker. If `-usr` is defined without `-pw`, the user is prompted for the password at runtime.

### **-pw *password***

(Required, unless `-prompt` is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the `-prompt` option.

### **-rp *repositorypath***

(Required) Specifies the repository path.

### Example: Delete Item or Path

To delete the item `TestItem` and path `TestPath` from the repository path `\TestRep\TestPath`:

```
hrepmgr -b brokername -usr username -pw password -del TestItem TestPath -rp \TestRep\TestPath
```

## Rename Item or Path

The option `-ren`, is used with `hrepmgr` to rename an item or repository path, and has the following syntax:

```
hrepmgr -ren -oldname name -newname name -rp path {options}
```

### **-ren**

(Required) Specifies to rename a repository item, or path. The options `-rp`, `-oldname`, and `-newname` must be used with this option.

### **-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-rp *repositorypath***

(Required) Specifies the repository path.

**-oldname *name***

(Required) Specifies the name of the repository item, or path to rename.

**-newname *name***

(Required) Specifies the new name of the repository item, or path to rename.

**-isv**

(Optional) Specifies to ignore the snapshot view.

**Example: Rename Item or Path**

To rename path TestPath1 of the repository path \Rep\Path to TestPath2 without checking the snapshot view:

```
hrepmgr -b brokername -usr username -pw password -ren -rp \Rep\Path -oldname  
TestPath1 -newname TestPath2 -isv
```

## Update Item or Path Property

The option -upd is used with hrepmgr to update Item or Path property, and has the following syntax:

```
hrepmgr itemname(s) or pathname(s) or "pattern" -upd -rp path {options}
```

**-upd**

(Required) Specifies to update a repository, item, or path.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-rp *repositorypath***

(Required) Specifies the repository path.

**-addvgrp *groupname(s)***

(Optional) Specifies to add view group(s).

**Note:** Options for adding and removing group(s) for view access can be used together.

**-remvgrp *groupname(s)***

(Optional) Specifies to remove view group.

**Note:** Options for adding and removing group(s) for view access can be used together.

**Example: Update Item or Path Property**

To add group TestGrp1 and TestGrp2 to view access, and remove group TestGrp3 and TestGrp4 from view access for item TestItem and path TestPath from the repository path \TestRep\Path:

```
hrepmgr -b brokername -usr username -pw password -upd TestItem  
TestPath -rp \TestRep\Path -addvgrp Grp1 Grp2 -remvgrp Grp3 Grp4
```

## Create Empty Path

The option -cr is used with hrepmgr to create an empty path, and has the following syntax:

```
hrepmgr -cr -rp path {options}
```

**-cr**

(Required) Specifies to create the repository or path.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-rp *name* | *repositorypath***

(Required) Specifies the repository path.

**-all**

(Optional) Specifies all baseline, an option for create empty path. This option creates the path structure in all existing baselines of the repository.

**Example: Create Empty Path**

To create an empty path TestPath2 under \TestRep\TestPath1, and update the repository only with the new item path:

```
hrepmgr -b brokername -usr username -pw password -cr -rp  
\TestRep\TestPath1\TestPath2
```

## hri Command-Remove Item

The hri command executes the remove item process specified on the command line, or the first remove item process in the current context if no process name is specified. The process definition determines the default behavior and the process access. Other aspects of the process can be specified through command parameters.

This command has the following format:

```
hri item_names {-b name -en name -st name} {-usr username -pw password} {-vp path}  
[-pn name] [-ot | -ob] [-p name] [-de string] [-v] [-prompt] [-i inputfile.txt  
| -di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

***item\_names***

Specifies the names of the items to be removed.

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *name***

(Required) Specifies the name of the state in which the user is operating.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies Defines the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-vp *path***

(Required) Specifies the CA Harvest SCM view path that corresponds to the current working directory.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-ot**

(Optional) Specifies that the new remove item version is to be created on the trunk. This parameter is mutually exclusive with -ob.

**Note:** This option is only available if the selected or default remove item process allows this operation to be applied on the trunk.

**-ob**

(Optional) Specifies that the new remove item version is to be created on a branch. This parameter is mutually exclusive with -ot.

**Note:** This option is only available if the selected or default remove item process allows this operation to be applied on a branch.

**-p *name***

(Optional) Specifies the package name, read from the command line or a signature file.

**-de *string***

(Optional) Specifies a comment to be associated with the removed versions of items created during the process execution.

**UNIX and Linux:** If this option is not specified, you are prompted to enter a description. You can enter multiple lines of description in prompt mode. To terminate the description, enter a period (.) and a carriage return on an otherwise blank line or use an end of file marker Ctrl+D on a blank line.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hrmvpth Command-Remove Path

The hrmvpth command executes the remove path process specified on the command line, or the first remove path process in the current context if no process name is specified. The process definition determines the default behavior and the process access. Other aspects of the process can be specified through command parameters.

This command has the following format:

```
hrmvpth {-b name -en name -st name -vp path -np path} {-p name} {-usr username -pw  
password} [-pn name] [-ot | -ob] [-de description] [-prompt] [-i inputfile.txt | -di  
inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Optional) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *name***

(Required) Specifies the name of the state in which the user is operating.

**-vp *path***

(Required) Specifies the CA Harvest SCM view path that is to be removed.

**-p *name***

(Required) Specifies the CA Harvest SCM package name, read from the command line or a signature file.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at run time.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-ot**

(Optional.) Specifies that the new remove path version and all of its subitem versions are to be created on the trunk. This parameter is mutually exclusive with -ob.

**Note:** This option is only available if the selected or default remove path process allows this operation to be applied on the trunk.

**-ob**

(Optional) Specifies that the new remove path version and all of its subitem versions are to be created on a branch. This parameter is mutually exclusive with -ot.

**Note:** This option is only available if the selected or default remove path process allows this operation to be applied on a branch.

**-de *description***

(Optional) Specifies a description of the new version created during the remove path process. The description is stored in the database.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.



## hrnitm Command-Rename Item

The `hrnitm` command executes the rename item process specified on the command line, or the first rename item process in the current context if no process name is specified. The process definition determines the default behavior and the process access. Other aspects of the process can be specified through command parameters.

This command has the following format:

```
hrnitm {-b name -en name -st name -vp path -on name -nn name} {-p name} {-usr username  
-pw password} [-pn name] [-ot | -ob] [-ur] [-uk] [-de description] [-prompt]  
[-i inputfile.txt | -di inputfile.txt] [-eh filename] [-o filename | -oa filename]  
[-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Optional) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *name***

(Required) Specifies the name of the state in which the user is operating.

**-vp *path***

(Required) Specifies the CA Harvest SCM view path that corresponds to the current working directory.

**-on *name***

(Required) Specifies the name of the file that is being renamed.

**-nn *name***

(Required) Specifies the new name of the file that is being renamed.

**-p *name***

(Required) Specifies the CA Harvest SCM package name, read from the command line or a signature file.

**-usr *username***

(Required, unless `-prompt` is specified) Specifies the user name for login to the broker. If `-usr` is defined without `-pw`, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless `-prompt` is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the `-prompt` option.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-ot**

(Optional) Specifies that the new rename item version is to be created on the trunk. This parameter is mutually exclusive with **-ob**.

**Note:** This option is only available if the selected or default rename item process allows this operation to be applied on the trunk.

**-ob**

(Optional) Specifies that the new rename item version is to be created on a branch. This parameter is mutually exclusive with **-ot**.

**Note:** This option is only available if the selected or default rename item process allows this operation to be applied on a branch.

**-ur**

(Optional) Creates a normal version for the rename item change. If a reserved version already exists in the same package, it is updated as a normal version.

**-uk**

(Optional) Creates a normal version for the rename item change. Another new reserved version is created after the normal version.

**-de *description***

(Optional) Specifies a description of the new version created during the rename item process. The description is stored in the database.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hrnpth Command-Rename Path

The hrnpth command executes the rename path process specified on the command line, or the first rename path process in the current context if no process name is specified. The process definition determines the default behavior and the process access. Other aspects of the process can be specified through command parameters.

This command has the following format:

```
hrnpth {-b name -en name -st name -vp path -on name -nn name} {-p name} {-usr username  
-pw password} [-pn name] [-ot | -ob] [-ur] [-uk] [-de description] [-prompt] [-i  
inputfile.txt | -di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg]  
[-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Optional) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *name***

(Required) Specifies the name of the state in which the user is operating.

**-vp *path***

(Required) Specifies the CA Harvest SCM view path that is to be renamed.

**-nn *name***

(Required) Specifies the new name of the path that is being renamed.

**-p *name***

(Required) Specifies the CA Harvest SCM package name, read from the command line or a signature file.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-ot**

(Optional) Specifies that the new rename path version and all its subitem versions are to be created on the trunk. This parameter is mutually exclusive with -ob.

**Note:** This option is only available if the selected or default rename path process allows this operation to be applied on the trunk.

### **-ob**

(Optional) Specifies that the new rename path version and all its subitem versions are to be created on a branch. This parameter is mutually exclusive with `-ot`.

**Note:** This option is only available if the selected or default rename path process allows this operation to be applied on a branch.

### **-ur**

(Optional) Creates a normal version for the rename path change. If a reserved version already exists in the same package, it is updated as a normal version.

### **-uk**

(Optional) Creates a normal version for the rename path change. Another new reserved version is created after the normal version.

### **-de *description***

(Optional) Specifies a description of the new version created during the rename path process. The description is stored in the database.

### **-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the `svrenc` utility.

**Note:** `-eh` overrides `-usr` and `-pw` options. `-prompt` overrides `-eh`, `-usr` and `-pw` options.

## hrt Command-Restore

The CA Harvest SCM restore utility (`hrt`) lets administrators move archived data back to its original database.

**Note:** Administrators can use the archive (`har`) utility to move data from the CA Harvest SCM database tables to another location.

The following rules apply to the restore function:

- A CA Harvest SCM broker and server must be running on the CA Harvest SCM installation on which you want to run the restore.
- Only CA Harvest SCM administrators can run the restore utility.
- An archived project can only be restored to its original database.
- An archived project can only be restored to the same version of CA Harvest SCM that was used to archive it.

**Important!** Projects archived prior to Version 12.0 need to be restored to the prior release before you upgrade to Version 12.0. However, this limitation will be removed after Version 12.0. Projects archived from Version 12.0 do not have to be restored prior to future upgrades/releases.

- In one restore execution, you can restore single or multiple projects from one archive file. You cannot restore multiple projects from different archives in one execution.
- The restoration will fail if the archive files (.arv, .dat) have been modified.
- The archive files (.arv, .dat) must reside together.
- You must specify the full path to the archived files.

After a successful restoration, the project status is automatically set to Inactive.

This command has the following format:

```
hrt {-b name} {-f} [-m computername -rport portnumber] [-musr username -mpw password]
{-usr username -pw password} [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh
filename] [-er filename] [-o filename | -oa filename] [-arg] [-v] [-wts] [-h]
project_name
```

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-f**

(Required) Specifies the archive file.

**-m computername**

(Optional) Specifies the name of the destination computer for the archive, if you are archiving to or restoring from a remote computer.

**-rport portnumber**

(Optional) Specifies the port number of the destination computer for the archive, if you are archiving to or restoring from a remote computer.

**-musr *username***

(Optional) Specifies the username of the destination computer for the archive, if you are archiving to or restoring from a remote computer.

**-mpw *username***

(Optional) Specifies the password of the destination computer for the archive, if you are archiving to or restoring from a remote computer.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-er *filename***

(Optional) Specifies the encrypted remote username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility. -er overrides -musr and -mpw options.

**More information:**

[har Command-Archive](#) (see page 20)

## hsigget Command-Signature File Query

The hsigget command retrieves and displays the information stored in CA Harvest SCM signature files. You can query for information on one or more specific files. You can also request information for all attributes or query specific attributes by name. If no attributes are specified, all of them are reported.

The output from the hsigget command returns a status for each file found. The status codes are:

- Untouched
- Modified
- Removed

**Note:** If the signature file cannot be opened, this command returns **-1**. If you renamed the signature file, it is considered as having been removed or deleted and the command returns **3**.

This command has the following format:

```
hsigget filename... [-a attribute1 attribute2...] [-context] [-purge {[filename...]
| all}] [-v] [-t] [-o filename | -oa filename] [-i inputfile.txt | -di inputfile.txt]
[-arg] [-gl] [-wts] [-h]
```

***filename...***

Specifies one or more file names in the signature file to query.

***-a attribute1 attribute2...***

(Optional) Specifies one or more attributes to be reported.

The following attributes can be queried and modified with the hsigget command. They are identified by their symbolic name.

**author**

Specifies the user who created the reserved version for a check out for update, or the user who created the version through check-in. The value of this attribute is the real name supplied in the User Properties, if it exists; otherwise, it is the user name. For check out for update, the author and user are the same. The value of this attribute is updated during:

- Check-out (reserve only)
- All modes of check-in

**ctime**

Specifies the time a version was created. For a reserved version, this is set either through check out for update or check in for update and keep. For check out for browse, this is version creation time.

The value of this attribute is updated during:

- Check-out (except for reserve only)
- All modes of check-in

#### **environment**

Specifies the project context of the corresponding item.

The value of this attribute is updated during:

- Check-out (except for reserve only)
- All modes of check-in

#### **modtime**

Specifies the modification time of the file when the signature file was last updated by check-in or check-out. Used to determine if the file has been modified since it was checked out or in.

The value of this attribute is updated during:

- Check-out (except for reserve only)
- All modes of check-in

#### **package**

Specifies the package name used during a check out for update, blank if checked out for browse.

The value of this attribute is updated during:

- Check out for update
- Check in for update and release
- Check in for release only (clears value)

#### **size**

Specifies the file size in bytes when the signature file was last updated. Used to determine if the file has been modified since it was checked out or in.

The value of this attribute is updated during:

- Check-out (except for reserve only)
- Check-in

#### **state**

Specifies the state context of the corresponding item.

The value of this attribute is updated during:

- Check-out (except for reserve only)
- Check-in (except for release only)



**status**

Calculates if the file has been modified since it was last checked in or out. This attribute is calculated when its value is queried by comparing the modtime and size attributes with the actual file modification time and size. Possible values are Modified, Untouched, or Removed.

The value of this attribute is calculated when queried.

**tag**

Specifies if the version is tagged as reserved. An empty string represents no tag.

The value of this attribute is updated during:

- Check-out (except for reserve only)
- Check in for update and release
- Check in for release only (removes reserved tag)

**user**

Specifies the user who performed the check-out. The value of this attribute is the real name supplied in the User Properties, if it exists; otherwise it is the user name.

The value of this attribute is updated during:

- Check-out (except for reserve only)
- All modes of check-in

**version**

Specifies the environment version number of the corresponding item.

The value of this attribute is updated during:

- Check-out (except for reserve only)
- Check in for update and release, if differences are found
- Check in for update and keep, if differences are found

**versionid**

Specifies a number that uniquely identifies the version, updated on check-in and check-out, and used as a basis for the synchronize process.

The value of this attribute is updated during:

- Check-out (except for reserve only)
- All modes of check-in

**viewpath**

Specifies the view path context of the corresponding item.

The value of this attribute is updated during:

- Check-out (except for reserve only)
- All modes of check-in

**-context**

Specifies to report information about the context variables in the signature file set by hsigset.

**-purge [filename...] | all**

(Optional) Specifies to remove information about files that are no longer located in the current directory. If the -purge option is followed by one or more optional file names, information for only those file names is removed. If the -purge option is followed by all, information for all files is removed.

**-v**

(Optional) Specifies to generate a report with a pre-selected subset of attributes related to version information, including file name, status, tag, version, and package.

**-t**

(Optional) Specifies that the output should be specially formatted to make it easy to import into a spreadsheet or word processor. Each field in the output is separated with a single tab.

Fields that contain multiple lines of data present a special challenge for report generation. Because new-line characters separate records, if a field also contains new-line characters, the output becomes ambiguous and difficult to format.

To make it easy to distinguish between a field separator and a record separator, special processing is done on multiple line output. Multiple line output for one field is merged into one long line with the string <NL> substituted for any occurrence of a new-line. Any occurrence of tabs in the data is replaced with the string <TAB>.

This algorithm removes any ambiguity between field and record separators. After the data is imported into the spreadsheet or table, it is easy to reformat by searching and replacing the <NL> and <TAB> strings.

When -t is not specified, each attribute of a record is specified on a new line and two new-lines separate records. If -nh is not specified, the output value follows the field name, unless the output is multi-line; in which case, output starts on a new line.

**-gl**

(Optional) Specifies that the time format is output according to the local settings.

#### Examples: hsigset command

To report attribute information for “package” and “environment” that is contained in the signature file for the file `foo.cpp`, use the following command:

```
hsigset foo.cpp -a package environment
```

To report version attributes contained in the signature file, use the following command:

```
hsigset -v
```

To report information about the context variables in the signature file set by `hsigset`, use the following command:

```
hsigset -context
```

To remove information about all file names contained in the signature file, use the following command:

```
hsigset -purge all
```

To remove information that is contained in the signature file for file `foo.cpp`, use the following command:

```
hsigset -purge foo.cpp
```

## hsigset Command-Signature File Update

The `hsigset` command sets the context attributes to be used in the `hci` and `hco` command lines and assigns attributes to a directory. If command line options are not present and no information for the item or file exists in the signature file, these attributes are used during command line check-in and check-out as defaults.

If the CA Harvest SCM signature file cannot be opened, the command line returns **-1**.

This command has the following format:

```
hsigset . [-context attribute=value] [-purge context attribute...] all ]  
[-i inputfile.txt | -di inputfile.txt] [-o filename | -oa filename] [-h]
```

**Note:** A period (.) in the syntax directly following the command is used to refer to the current directory.

**-context *attribute*="value"**

(Optional) Defines the values for the context attribute using an attribute symbolic name. You can define an unlimited number of context attributes.

Valid attributes are: author, crtime, environment, modtime, package, size, state, status, tag, user, version, versionid, and viewpath.

For descriptions of these attributes, see the -a option of hsigset Command-Signature File Query.

**-purge context *attribute*...**

(Optional) Specifies to remove a given context attribute from the signature file. If the -purge option is followed by an attribute name, information only for that attribute name is removed.

Valid attributes are: author, crtime, environment, modtime, package, size, state, status, tag, user, version, versionid, and viewpath.

**Note:** For descriptions of the attributes, see the -a option of hsigset Command-Signature File Query.

**Example: hsigset command**

To set context attribute information for "environment" to "My environment" and the state to "Dev," use the following command:

```
hsigset -context environment="My environment" state=Dev
```

To set context attribute information for the view path to "abc," use the following command:

```
hsigset -context viewpath="abc"
```

To remove information for "environment" and "state," use the following command:

```
hsigset -purge environment state
```

To remove information about all context variables contained in the signature file, use the following command:

```
hsigset -purge all
```

## hsmtp Command-SMTP Mail Utility

The hsmtp mail command lets you use Simple Mail Transfer Protocol (SMTP) for sending email messages directly from the command prompt. Many email systems that send mail over the internet use SMTP to send messages from one server to another; the messages can then be retrieved with an email client using either Post Office Protocol (POP) or Internet Message Access Protocol (IMAP). SMTP is generally used to send messages from a mail client to a mail server.

To set the following mail options, use the command line or create an hsmtp.arg file in the %CA\_SCM\_HOME% directory. Options set on the command line override those specified in the hsmtp.arg file. If no options are specified in the hsmtp.arg file or on the command line, the defaults are used.

**Note:** For more information about the hsmtp.arg file, see the *Administrator Guide*.

This command has the following format:

```
hsmtp.exe {-m server_name | IP_address} [-p port_number] {-f address} [-s  
subject_line] [-d] [-cc address] [-bcc address] [-arg] [-wts] [-h]
```

**-m server\_name | IP\_address**

(Required, if not specified in the hsmtp.arg file.) Specifies the name or IP address of the SMTP mail server that should be used to send the email.

**-p port\_number**

(Optional) Specifies the port number. By default, SMTP listens on port 25. You can use the -p option to change the default port.

**-f address**

(Required, if not specified in the hsmtp.arg file.) Specifies the sender's email address.

**-s subject\_line**

(Optional) Specifies an individual subject line in the email. If not specified, hsmtp uses the default subject: "Message from CA Software Change Manager."

**-d**

(Optional) Prints debugging information to the standard output device.

**-cc address**

(Optional) Specifies email addresses of persons who should receive a carbon copy of the email.

**-bcc address**

(Optional) Specifies email addresses of persons who should receive a blind carbon copy of the email.

#### Example: hsmtmp command

```
hsmtmp.exe -m smtpmail.company.com -p 25 -f sender@company.com  
"recipient1@company.com"
```

## Send an Email

To send an email, enter the hsmtmp command and desired options at the command line.

#### To send an email

1. Enter the hsmtmp command, specifying the options you want, and then press Enter.  
The command line starts text entry mode.
2. Enter the text of your email. You can enter as many lines as you want, pressing Enter at the end of each line. Then enter the end-of-file marker:

**Windows:** Ctrl+Z

**UNIX:** Ctrl+D

3. Press Enter again.  
The email is sent.

## hspp Command-Switch Package

The switch package command lets you move versions from a source package to a target package. The following rules apply to the switch package command:

- Versions can be partially switched to another package. For example, package A has item1 (version 1), item2 (version 2), and you can switch item1 (version1) to a different package and still retain item2 (version2).
- You can only switch to a package that is located in the same state as the original package.
- You must follow the CA Harvest SCM branch-trunk version rule: one branch belongs to only one package; one package cannot have multiple versions of an item that exists in either branch-trunk or branch-branch.

After the process is defined in the project life cycle, users are able to execute this process from the command line. The CA Harvest SCM Administrator can add the switch package process to the various states in a project life cycle by using the Administrator application or CA Harvest SCM Web Interface.

This command has the following format:

```
hspp {-fp name -tp name} {-b name -en name -st name} [-pn name] {-usr username -pw  
password} [-s filename | pattern] [-v] [-prompt] [-i inputfile.txt  
| -di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-fp name**

(Required) Specifies the source (from) package name.

**-tp name**

(Required) Specifies the target (to) package name.

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-en name**

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, the name should be enclosed by quotation marks.

**-st name**

(Required) Specifies the name of the state in which the user is operating.

**-pn name**

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-usr username**

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-s *filename* | *pattern***

(Optional) Specifies versions to be switched from the source package to the target package. If this option is not present, then all versions from the source package are moved to the target package.

This option must be followed by a filename or a wildcard pattern (for example, \*.c).

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**Examples: hspp command**

In the following examples, packageA is the source package and packageB is the target package.

To move all versions in packageA to packageB:

```
hspp -fp packageA -tp packageB required options...
```

or

```
hspp -fp packageA -tp packageB required options... -s "*.c"
```

To move all versions of files with an extension of .c in packageA to packageB:

```
hspp -fp packageA -tp packageB required options... -s "*.c"
```

To move all versions of files foo.c in packageA to packageB:

```
hspp -fp packageA -tp packageB required options... -s "foo.c"
```



To move all versions of files foo.c and bar.c in packageA to packageB:

```
hspp -fp packageA -tp packageB required options... -s "foo.c" "bar.c"
```

To move version 1.0 of file foo.c in packageA to packageB:

```
hspp -fp packageA -tp packageB required options... -s "foo.c:1.0"
```

To move versions 1.0 and 2.0 of file foo.c in packageA to packageB:

```
hspp -fp packageA -tp packageB required options... -s "foo.c:1.0" "foo.c:2.0"
```

## hsql Command-RDB Query

The hsql relational database query utility enables you to generate reports from a client computer. This utility is limited to executing SQL SELECT queries, and cannot alter the contents of the CA Harvest SCM database. Output is generated in a form that can be imported into spreadsheet or word processing programs. If no input and output options are specified, hsql reads input from the standard input and writes output to the standard output.

When hsql is executed, from the command line or a user-defined process (UDP), a server process must be running to transact with the relational database.

When using the CA Harvest SCM hsql relational database query command line in the Program field of the UDP Properties dialog, the client and server UDP behaviors differ.

**Note:** Your SQL input cannot begin with a comment; it must begin with a SELECT statement.

This command has the following format:

```
hsql {-b name} [-f name] {-usr username -pw password} [-prompt] [-i inputfile.txt |  
-di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-nh] [-s] [-t] [-gl]  
[-arg] [-wts] [-h]
```

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-f name**

(Optional) Specifies an input file name. If a full path is not specified for the file, it must exist in the directory where the command is invoked. The -f option can be omitted on platforms that support the standard input, as most UNIX and Linux platforms do. In this case, the standard input is read for the SQL statement.

The input file contains the SQL commands to be executed by hsql. Only query statements that start with the keyword SELECT are executed by hsql.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-nh**

(Optional) Specifies that a header should not be included in the command output. By default, the header is included in the output. The header is generated based on the table column names being queried, or as specified in the SQL being executed.

**-s**

(Optional) Specifies to suppress the prompt that is displayed when no input file name is provided. When -s is not specified and no input file is specified, the following prompt is displayed:

Enter SQL command:

**UNIX and Linux:** End input by typing at the start of a new-line by typing Ctrl-D and then Enter.

**Windows:** End input by typing at the start of a new-line by typing Ctrl-Z and then Enter.

The display of this prompt can be undesirable in some situations. For example, if you plan to redirect the output of hsql to be used by another process, do not include this prompt.

**-t**

(Optional) Specifies the formatting of output when using the hsigget or hsql commands. When -t is specified, the output is specially formatted to make it easy to import into a spreadsheet or word processor. Each field in the output is separated with a single tab.

Fields that contain multiple lines of data present a special challenge for report generation. Because new-line characters separate records, if a field also contains new-line characters, the output becomes ambiguous and difficult to format.

To make it easy to distinguish between a field separator and a record separator, special processing is done on multiple line output. Multiple line output for one field is merged into one long line with the string <NL> substituted for any occurrence of a new-line. Any occurrence of tabs in the data is replaced with the string <TAB>.

This algorithm removes any ambiguity between field and record separators. After the data is imported into the spreadsheet or table, it is easy to reformat by searching and replacing the <NL> and <TAB> strings.

When -t is not specified, each attribute of a record is specified on a new line and two new-lines separate records. If -nh is not specified, the output value follows the field name, unless the output is multi-line; in which case, output starts on a new line.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-gl**

(Optional) Specifies that the time format will be output according to the local settings.

**-l**

Converts the time in the local time zone.

## hsv Command-Select Version

The hsv command selects item versions from the specified view path. The various options work in conjunction with one another to allow you to select versions according to multiple criteria.

This command has the following format:

```
hsv {-b name -en name -vp path} {-usr username -pw password} [-st name] [-p name] [-iu  
creator_name] [-io option] [-iv option] [-it option] [-ib option] [-id option] [-s  
filename | pattern] [-v] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh  
filename] [-o filename | -oa filename] [-gl] [-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-vp *path***

(Required) Specifies the CA Harvest SCM view path that corresponds to the current working directory.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-st *name***

(Optional) Specifies the name of the state in which the user is operating.

**-p *name***

(Optional) Specifies the CA Harvest SCM package name, read from the command line or a signature file.

**-iu *creator\_name***

(Optional) Specifies version selection based on the name of the user who created the version. Specify a user name for filtering.

When the Item User option is used with the Latest in View Version filter, Select Version (hsv) will list the latest version in that view only if the specified user created it; if the user did not create this version, no version is listed. When hsv is set to Latest, the latest version created by the user is displayed. For All and All in View, only the versions created by the specified user that match the other filtering criteria are listed.

**-io *option***

(Optional) Specifies version selection based on the modification status of items. Available options are:

**a**

**All**-Selects versions for all items regardless of whether they have been modified.

**m**

**Modified**-Selects only versions for items that have been modified in this project.

For some processes, such as delete version and merges, initial versions cannot be used. Selecting Modified for this filter will exclude all items for which only initial versions exist. This filter is also useful if you are checking out only changed items and want to prevent unchanged ones from being displayed.

**nm**

**Not Modified**-Selects versions for items only if the item has not been modified in this project. This is the initial version, or version 0, of items for which no other versions exist.

**-iv option**

(Optional) Specifies what type of versions to select based on the view. Available options are:

**lv**

**Latest in View**-Selects the latest for each item in the current view that match the other filtering criteria. Only one version per item is displayed. This option is incompatible with viewing branch versions because they are not in a view. When selected, the only Branch option available is Trunk Only.

**av**

**All in View**-Selects all versions matching the other filtering criteria that exist in the view associated with the current state. Any later versions of an item in another view that have not yet been promoted to the current state are not displayed. This option is incompatible with viewing branch versions because they are not in a view. When selected, the only Branch option available is Trunk Only.

**a**

**All**-Selects all versions, including branch versions and trunk versions that do not yet exist in the view associated with the current state, that match the other filtering criteria. This option is not available when selecting versions in snapshot views.

**l**

**Latest**-Selects the latest for each item that match the other filtering criteria, including those that do not yet exist in the view associated with the current state. Only one version per item is displayed. This option is not available when selecting versions in snapshot views.

**-it option**

(Optional) Specifies version selection based on the tag status of versions. Available options are:

**nt**

**No Tag**-Selects only versions that do not have a tag associated with them and match the other filtering criteria.

**r**

**Reserved**-Selects only versions tagged as reserved (R) that match the other filtering criteria.

**m**

**Merged**-Selects only versions tagged as merged (M) that match the other filtering criteria.

**rm**

**Removed**-Selects only versions tagged as removed (D) that match the other filtering criteria.

**at**

**Any**-Selects only versions that have been tagged as reserved (R), merged (M), or removed (D), and match the other filtering criteria.

**-ib option**

(Optional) Specifies version selection based on the location of the version on the trunk and branch. Available options are:

**to**

**Trunk Only**-Selects versions on the trunk in the project specified that match the other filtering criteria.

**b**

**Branch Only**-Selects all branch versions that match the other filtering criteria. This option is incompatible with the All in View or Latest in View Version filters because branches do not exist in a view.

**tb**

**Trunk and Branch**-Selects all versions that match the other filtering criteria.

**ub**

**Unmerged Branch**-Selects all unmerged branch versions that match the other filtering criteria. This option is incompatible with the All in View or Latest in View Version filters because branches do not exist in a view. Use this option to select versions before executing the concurrent merge process.

**-id option**

(Optional) Specifies a date range for the selection. Available options are:

**a**

**All**-All files are selected, indicating date checking will not be performed.

**sd**

**Specified Date**-Selects versions specified by date From and date To. The date format is *mm/dd/yyyy*.

**Example: sd option**

`-id sd 03/14/95 10/26/97`

**nd**

**Previous Date**-Selects previous x number of days.

**-s filename | pattern**

(Optional) Specifies that the check-in process should search every directory below the current one for files that match a file name or naming pattern. This option must be followed by a file name or naming pattern. To specify multiple file names or naming patterns, use the -s option before the first file name or pattern. -s searches for the patterns specified between it and the next option.

Naming patterns that include a wildcard must be enclosed in single or double quotes when used with the -s option. If no quotes are included on multiple naming patterns, the process searches recursively for the first file (the naming pattern immediately following the -s option) and searches only the current directory for the remaining files.

#### Examples: -s option

To recursively select all versions with the .c file extension, use the following command:

```
hsv options -s "*.c"
```

To select versions \*.c and \*.exe and recursively select versions one.c, two.c, and three.c, use the following command:

```
hsv file.txt *.c ... -s one.c two.c three.c options *.exe.
```

**-eh filename**

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**-gl**

(Optional) Specifies that the time format will be output according to the local settings.

## hsync Command-Synchronize Directory from View Path

The hsync command updates a file system directory so it reflects updates to the contents of a CA Harvest SCM view under a specified path.

**Note:** For descriptions of key concepts for understanding the hsync command, see the *Administrator Guide*. This guide defines terminology that the hsync command information uses.

This command has the following format:

```
hsync {-b broker} {-en project} {-st state} {-vp viewpath} {-cp clientpath}  
{-usr username -pw password | -eh filename | -prompt} [-pn processname] [-br | -sy]  
[-av | -fv] [-iv | -pl packagelist | -il itemlist | -iol | -ps | -pv] [-ss snapshotview  
] [-bo | -to | -tb] [-ced] [-rm name -rport portnumber -rusr username -rpw password  
| -er encryptedpwfile] [-purge | -complete] [-excl names] [-excls patterns] [-v] [-i  
inputfile | -di inputfile] [-o logfile | -oa logfile] [-wts] [-h]
```

**-b *broker***

(Required) Specifies the broker name.

**-en *project***

(Required) Specifies the project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *state***

(Required) Specifies the name of the state in which the user is operating.

**-vp *viewpath***

(Required) Specifies the CA Harvest SCM view path under which items and versions are selected.

**-cp *clientpath***

(Required) Specifies the client path that corresponds to the current view path. In hsync terms, this path is the reference directory. It must be an absolute path.

**-usr *username***

(Required, unless -prompt is specified) Defines the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Defines the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.



**-pn *processname***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first executable check-out process allowing the specified check-out mode in the specified state is invoked.

**-br**

(Optional) Specifies that the check-out process is to be executed in Browse mode. The specified check-out process must allow Browse mode.

**Note:** -br and -sy are mutually exclusive.

**-sy**

(Optional) Specifies that the check-out process is to be executed in Synchronize mode. This is the default setting for check-out mode. The specified check-out process must allow this mode.

**Note:** -sy and -br are mutually exclusive.

**-av**

(Optional) Sets the reference directory type to Active View.

**Note:** -av and -fv are mutually exclusive.

**-fv**

(Optional) Sets the reference directory type to Full View. This is the default setting for reference directory type.

**Note:** -fv and -av are mutually exclusive.

**-iv**

(Optional) Sets item selection option to All Items in View. This is the default setting for item selection. Item selection options are described in How hsync Works earlier in this chapter.

**Note:** -iv and -pl, -il, -ps, -pv are mutually exclusive.

**-pl *packagelist***

(Optional) Specifies the package list for item selection. One or more package names must follow this option. You can specify the ["package"] system variable when hsync is executed in a UDP where the package variable is available.

**Note:** The hsync command does not consider the provided package list for deciding which version should be present in the client location. The hsync role is limited to selecting the items in the list. Items which are present under any of the packages present in the package list are selected for synchronization. The latest versions per the timestamps of those items are checked out to the clientpath.

### Example: Use -pl to Synchronize the View Path With the File System Directory

This example demonstrates how to synchronize the view path with the file system directory using the item selection packagelist option (-pl).

Packages p1, p2, and p3 have versions 0.1.1, 0.2.1, and 0.3.1, respectively, for an item File1.txt. Version 0.3.1 has the latest time stamp.

Use the following command:

```
hsync -b brokername -en projectname -st statename -vp viewpath -cp clientpath -tb  
-pl p1 p2 -usr username -pw password
```

This command updates the file system directory with version 0.3.1 of File1.txt.

**Note:** Although the package p3 is not included in the package list, the version from this package is selected because it is the latest version per the timestamp.

#### **-il *itemlist***

(Optional) Sets item selection option to Item List. One or more item names must follow this option. The item names must include the item path. The path may be absolute or relative to the given view path. If an absolute path is specified, hsync ignores any item that does not begin with the given view path.

#### **-iol**

(Optional) Supplements the item selection -il option. For every version in the process' list of item and path versions, the itemobjid of the version is used for the itemobjid system variable.

**Note:** To support use with the ["version"] system variable, items may have the version number appended. These version numbers are ignored.

#### **-ps**

(Optional) Sets item selection option to All Packages in State.

#### **-pv**

(Optional) Sets item selection option to All Packages in View.

#### **-ss *snapshotview***

(Required if the specified state has a snapshot view; otherwise, this option is invalid.) Specifies the snapshot view name.

**Note:** -ss is mutually exclusive with -pl, -ps, -pv, and -av.

#### **-bo**

(Optional) Specifies to check out latest versions located on branches. If multiple versions exist in different packages for the same item, the version that is the latest per the timestamp is selected for check-out.

**Note:** -bo is mutually exclusive with -to, -tb, and -ss.

**-to**

(Optional) Specifies to check out latest versions located on the trunk. This option is the default.

**Note:** -to is mutually exclusive with -bo and -ss.

**-tb**

(Optional) Specifies to check out latest versions located on the trunk and branches. The version that has the latest timestamp is selected irrespective of its presence on the trunk or in any particular package on a branch.

**Note:** -tb is mutually exclusive with -bo, -to, and -ss.

**Example: Use -tb to Synchronize the View Path with the File System Directory**

This example demonstrates how to synchronize the view path with the file system directory using the trunk and branch option (-tb).

Packages p1 and p2 have versions 1 and 0.1.1, respectively, for an item File1.txt. Version 0.1.1 has the latest time stamp.

Use the following command:

```
hsync -b brokername -en projectname -st statename -vp viewpath -cp clientpath -tb  
-usr username -pw password
```

This command updates the file system directory with version 0.1.1 of File1.txt.

**-ced**

(Optional) Enables the creation of directories that contain no items, during synchronization. The default setting is to create no empty directories.

**-rm *name***

(Optional) Specifies the remote computer where a reference directory is located. Necessary only if the reference directory is not in the local file system. A CA Harvest SCM file agent must be running on the remote computer, and it must use the same rtserver as the specified broker.

**-rport *name***

(Optional) Specifies the port number of the remote computer. Specify this option only if a remote computer is specified.

**-rusr *username***

(Optional) Specifies the user name for the remote computer to check out from. Specify this option only if a remote computer is specified.

**-rpw *password***

(Optional) Specifies the password for the remote computer to check out from or check in to. Specify this option only if a remote computer is specified.

**-er**

(Optional) Specifies the encrypted remote username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -er overrides -rusr and -rpw options.

**-purge [filename...] | all**

(Optional) Specifies to delete files that are no longer located in the current directory and in subdirectories that have not been excluded and have no corresponding path version in the check-out list. If a directory meets those criteria and only contains a signature file after files are purged, then it is deleted. This is applicable only for read-only files. If a file is writable, then it is not deleted.

**-complete**

(Optional) Similar to -purge but deletes files that are writable in addition to the criteria specified in the -purge category.

**-excl list of filenames or directory names**

(Optional) This option can only be used with -purge and -complete. It lists explicit file and directory names to exclude from the purge or complete. To specify multiple names, use the -excl option before the first file name. -excl will exclude files and directories that match the names that are specified between it and the next option.

The hsync command treats a forward slash (/) and a backslash (\) as equivalent directory delimiters. (/ is preferable because \ is often interpreted as the beginning of an unprintable character). File name should be relative to the -cp argument.

Wildcards may be embedded in a file or directory name, but the name must be enclosed in quotes.

**-excls pattern**

(Optional) This option can only be used with -purge and -complete. It lists file name patterns to exclude from the purge or complete. It excludes files in any directory below the reference directory that match a pattern in the list. Naming patterns that include a wildcard must be enclosed in single or double quotes.

## htakess Command-Take Snapshot

The htakess command executes the take snapshot process specified on the command line, or the first take snapshot process in the current context if no process name is specified. The process definition determines the default behavior and the process access. Other aspects of the process can be specified through command parameters.

This command has the following format:

```
htakess {-b name -en name {-st name | -abv name} -ss name} {-usr username -pw password}  
[-p name] [-po] [-pb] [-vp path] [-pg name] [-ve] [-ts YYYY:MM:DD:HH:MM:SS:FR]  
[-rs name] [-v] [-pn name] [-prompt] [-i inputfile.txt | -di inputfile.txt]  
[-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-en name**

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st name**

(Required, unless -abv is specified.) Specifies the name of the state for the snapshot.

**Note:** This option is mutually exclusive with -abv.

**-abv name**

(Required, unless -st is specified.) Specifies an admin-based view, which takes a snapshot based on a view instead of a state. This operation involves no prelink or postlink processes.

**Note:** This option is mutually exclusive with -st.

**-ss name**

(Required) Specifies the name of the snapshot view to which the repository belongs. If used in the check-out process, the state specified must have All Snapshot Views enabled.

**-usr username**

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw password**

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-p *name***

(Optional) Specifies the CA Harvest SCM package name, read from the command line or a signature file.

**-po**

(Optional) Specifies that the latest versions in the specified packages (-p) are captured by the take snapshot process. If the -vp option is specified, the repository path structure (directory structure) is included in the snapshot along with any files that are both in the repository path and the package. Versions that are in the package, but are not in the view path (-vp option) will not be included in the snapshot.

**-pb**

(Optional) Specifies that the latest versions in the baseline plus the latest versions in the specified packages (-p) are captured by the take snapshot process. If the -vp option is used, then the repository path structure (directory structure) is included in the snapshot along with any files that are both in the repository path and the package.

**-vp *path***

(Optional) Specifies a repository path to include in the snapshot.

**-pg *name***

(Optional) Specifies a CA Harvest SCM package group.

**-ve**

(Optional) Specifies that the Visible to Other Projects option is enabled, which allows this snapshot to be listed on the Snapshot View dialog during the setup of the baseline view of other projects.

**-ts *YYYY:MM:DD:HH:MM:SS:FR***

(Optional) Specifies the Modified Version Date. The take snapshot process captures versions in the current working view that were modified before or on the specified version modified date and time. The snapshot time option cannot be used with the reference snapshot (-rs) option.

The following formats are also supported:

*YYYY-MM-DD HH:MM:SS:FR*

*YYYY:MM:DD HH:MM:SS:FR*

*YYYY,MM,DD HH,MM,SS,FR*

- The delimiter in the date and time can be '-',':',';',',', ' '.
- The delimiter between date and time can be '-',':',';',',', ' ' (space).

**-rs *name***

(Optional) Specifies the name of an existing snapshot to include in the take snapshot process. The package (-p) option, the package group (-pg) option, or both must follow the -rs option. The versions contained in the snapshot view specified by the -rs option, plus the packages specified in the -p, -pg, or both, are included in the new snapshot. The reference snapshot option cannot be used with the snapshot time (-ts) option.

**-pn *name***

(Optional) Specifies a specific process name to invoke if more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hucache Command-Manage Remote Site Cache

The hucache command executes remote site cache update and purge operations on demand. With hucache, latest trunk versions of a working view or snapshot view versions can be loaded into a specified cache location. Obsolete cache files can be purged using hucache by specifying an aging factor which determines which files should be deleted based on the last accessed date.

This command has the following format:

```
hucache {-b name} {-usr username -pw password} [-en name] [-st name | -ss name | -purge  
days] [-vp path] [-cacheagent name] [-rusr username] [-rpw password] [-eh filename]  
[-er filename] [-prompt] [-o filename | -oa filename] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the name of the project that will have its versions cached by this remote cache agent.

**-st *name***

(Required: st, -ss, and -purge are mutually exclusive and one is required.) Specifies the name of the state whose working view versions will be inserted into the cache. The latest trunk versions of all items in the view will be loaded into the cache.

**-ss *name***

(Required: -st, -ss, and -purge mutually exclusive and one is required.) Specifies the name of the snapshot view whose versions will be inserted into the cache. All item versions of the snapshot view will be loaded into the cache.

**-purge *days***

(Required: -st, -ss, and -purge are mutually exclusive and one is required.) Specifies an aging factor in days. Versions in the cache that have not been accessed for the specified number of days are purged out of cache to save disk space. Repositories belonging to the project specified by the -en parameter are processed. The aging of the file is calculated using the files "last accessed" date as maintained by the operating system.

**-vp *path***

(Optional) Specifies a repository path.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-cacheagent *computername***

(Optional) Specifies the remote cache agent host name.

**-rusr *username***

(Optional) Specifies the user name used to log in to the remote cache agent which is to have its cache updated or purged. This option is necessary only if you want to load a repository to a remote computer.

**-rpw *password***

(Optional) Specifies the password used to log in to the remote cache agent which is to have its cache updated or purged. This option is necessary only if you want to load a repository to a remote computer.

**-rport *portnumber***

(Optional) Specifies the remote cache agent port number.



**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides the -eh, -usr, and -pw options.

**-er *filename***

(Optional) Specifies the encrypted remote username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility. -er overrides -rusr and -rpw options.

## hudp Command-User-Defined Process

The user-defined process (UDP) lets you invoke an external program to run as a process in your life cycle. The program to execute, any command line parameters, and the options for its output were specified during definition of this process.

For programs that read from the default input device, default input parameters are specified. If this option is defined as modifiable in the process properties, these parameters can be overridden or modified at execution time. You can also supply additional command line parameters.

This command has the following format:

```

hudp {-b name -en name -st name} {-usr username -pw password} [-pn name] [-ip parameter]
[-ap parameter] [-v] [-prompt] [-i inputfile.txt | -di inputfile.txt] [-eh filename]
[-o filename | -oa filename] [-arg] [-wts] [-h]
  
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-st *name***

(Required) Specifies the name of the state in which the user is operating.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for the login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for the login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-pn *name***

(Optional) Defines a specific process name to invoke when more than one process of the same type exists in the state. If no process name is included in the command, the first process of that type in that state is invoked.

**-ip *parameter***

(Optional) Specifies input parameters. This option overwrites the original default parameter if that parameter is not secured.

**-ap *parameter***

(Optional) Specifies additional parameters. If you have more than one parameter, separate each of them by a space. Use double quotation marks around the parameters that contain spaces; otherwise, they are considered as two different parameters. For example, -ap "a 1", "a 2" a3 a4. Parameters that contain spaces are passed to the users process as a single argument along with the quotation marks. For example, "a 1" is passed to the users process as "a 1". Parameters that do not contain spaces are passed to the users process without the quotation marks even if you surround them with quotation marks. For example, "a3" and a4 are passed as a3 and a4.

If the administrator does not allow additional parameters, then using this option results in an error.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name that is used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## hup Command-Update Package

The hup command lets you update the package the -p option specifies. You can specify other aspects of the process through command parameters read from the command line. The various options work with each other to let you create a form, and add or delete form associations.

This command has the following format:

```
hup {-b name -en name -p name} {-usr username -pw password} [-npg name] [-at name]
[-pr number] [-af name] [-apg names...] [-rpg names...] [-des text] [-nt note] [-rf
name] [-ft formtype] [-afo formid] [-rfo formid] [-del] [-cf] [-v] [-prompt]
[-i inputfile.txt | -di inputfile.txt] [-eh filename] [-o filename | -oa filename]
[-arg] [-wts] [-h]
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-en *name***

(Required) Specifies the CA Harvest SCM project context. If the project name contains spaces, enclose the name in quotation marks.

**-p *name***

(Required) Specifies the CA Harvest SCM package name, read from the command line or a signature file.

**-usr *username***

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-npg *name***

(Optional) Specifies the new package name for an existing package.

**-at *name***

(Optional) Specifies the user name to assign to the package.

**-pr *number***

(Optional) Specifies the priority to assign to the package.

**-af *name***

(Optional) Specifies the name of a form to associate to the package. A form is created if it does not exist and if -cf is specified. The -ft option must be used in conjunction with this option.

**-apg *names...***

(Optional) Specifies package groups to add to the list of package groups associated with the current package.

**-rpg *names...***

(Optional) Specifies the package groups to disassociate from the current package. This does not delete the specified package group; it simply removes the association.

**-des *text***

(Optional) Specifies a package description.

**-nt *note***

(Optional) Specifies a note.

**-rf *name***

(Optional) Specifies the name of a form to disassociate from the package.

**-ft *formtype***

(Optional) Specifies the form type.

**-afo *formid***

(Optional) Specifies to add the form association specified by its object ID.

**-rfo *formid***

(Optional) Specifies to remove the form association specified by its object ID.

**-del**

(Optional) When removing a form (-rfo or -rf), if -del is specified and the form is not associated with a package, the form is deleted from the database. -del receives either the form type and name or formid from the -ft and -rf or -rfo options.

**-cf**

(Optional) Specifies to create a form by name if the form name does not exist. When added to -af (associate form), it creates a new form by name if the form name does not exist.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

## husrmgr Command-User Manager

The husrmgr command is a toolkit for the administrator who needs to maintain the user profile in CA Harvest SCM. The User Manager utility provides these user maintenance functions: import user, delete user, rename user, update user, and force change password.

- By default the program performs the import user mode.
- You can activate the delete user mode by specifying the -du (delete user) option.
- You can activate the update user mode by specifying the -ow (overwrite) option.
- You can activate the rename user mode by specifying the -nn (rename) option with -ow (overwrite) option.
- You can activate the change password on next login flag for all created and updated users by specifying the -cpw (change password) option.

The format of the data files depends on the combination of options used.

**Note:** An easy way to create the data file is by using Microsoft Excel. Enter the data in Excel and then save it to a file using the Save As, Text (Tab delimited) (\*.txt) option.

### Examples: husrmgr command

To create a user with username and password (required), define the user properties, and belonging to the user groups Developer, Development Manager, and Super User, use the following command:

```
JoeS<tab>MightyJoe<tab>Joe  
Smith<tab>(123)456-7890<tab><tab>(098)765-4321<tab>JOES@SOMEWHERE.COM<tab>Deve  
lop<tab>Development Manager<tab>Super User
```

To create a user with username and password (required), belonging to only the user group Public (automatic), use the following command:

```
JoeS<tab>MightyJoe
```

To create a user and add this user to the Developer group, use the following command:

```
JoeS<tab>MightyJoe<tab><tab><tab><tab><tab><tab><tab><tab>Developer
```

To clear fields in a user profile file, if you are using the following command:

```
husrmgr -b broker -usr username -pw password -ow filename
```

and the *filename* file contains the following:

```
john22<tab><tab>[clear]<tab>[clear]<tab>1234<tab><tab><tab><tab>
```

you get the following results for user john22:

- Password is unchanged.
- Real Name is cleared.
- Phone Number is cleared.
- Phone Extension is changed to 1234.
- Fax Number is unchanged.
- Email is unchanged.
- Note is unchanged.

If you are using the following command:

```
husrmgr -b broker -usr username -pw password -ow -cf [EMPTY] filename
```

and the *filename* file contains the following:

```
john22<tab><tab>[EMPTY]<tab>[EMPTY]<tab>1234<tab><tab><tab><tab>
```

you get the following results for user john22:

- Password is unchanged.
- Real Name is cleared.
- Phone Number is cleared.
- Phone Extension is changed to 1234.
- Fax Number is unchanged.
- Email is unchanged.
- Note is unchanged.

This command has the following format:

```
husrmgr {-b name} {-usr username -pw password} [-d\m symbol] [-ow [-nn]
[-cf clear_field]] [-du] [-cpw] [-v] [-prompt] [-i inputfile.txt | -di inputfile.txt]
[-eh filename] [-o filename | -oa filename] [-sw\ Y|N ] [-ad Y|N ] [-ae Y|N ] [-arg]
[-wts] [-h] filename
```

**-b name**

(Required) Specifies the CA Harvest SCM broker name.

**-usr username**

(Required, unless -prompt is specified) Specifies the user name for login to the broker. If -usr is defined without -pw, the user is prompted for the password at runtime.

**-pw password**

(Required, unless -prompt is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-d\m symbol**

(Optional) Specifies the symbol to use as the delimiter. The default symbol for delimiter is a tab.

**-ow**

(Optional) Specifies to activate the update user mode that enables administrators to overwrite existing user data. If the -ow option is not specified, existing user data will not change. If the password conflicts with current password policy, an error will occur.

**-nn**

(Optional) Specifies that user names can be changed. This parameter lets an administrator change the user name of an existing user object to another user name. This parameter must be used in conjunction with overwrite (-ow) mode.

**-cf clear\_field**

(Optional) Defines the literal used in the file format to clear a field's value. You can use this option to change the default literal used to clear a field's value. This option must be used in conjunction with the -ow option.

When updating user profiles, if a particular field value is provided in the input file, the corresponding field's value in the user profile is replaced with the new one from the file. If no value is provided for a particular field in the input file, the field's value remains unmodified in the user profile.

When the *clear\_field* literal is used as a value in the input file, the corresponding field's value is cleared in the user profile.

**Default:** [clear]

**Note:** The value of the -cf option is case-sensitive.

**-du**

(Optional) Specifies to activate the delete user mode that enables administrators to delete user data.

**-cpw**

(Optional) Specifies that all created and updated users must change their password on next login.

**-swl**

(Optional) Specifies whether the user account is restricted to running multiple CA Harvest SCM sessions from CA Harvest SCM clients to a single workstation for a given broker. Values for this switch are: Y=Restricted or N=Not Restricted.

**Default:** Not Restricted

**-ad**

(Optional) Specifies to disable a user account. Values for this switch are: Y=Disabled or N=Not Disabled.

**-ae**

(Optional) Specifies to force all created accounts to be marked as internal or external. Values for this switch are: Y=internal or N=external.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**More information:**

[husrmgr and Password Authentication](#) (see page 168)

## husrmgr and Password Authentication

When you are using internal authentication (CA Harvest SCM authentication), all user fields can be edited.

When you are using external authentication, such as Windows Active Directory, only the UserName and Note fields can be edited; no other fields can be edited (Password, RealName, Phone#, Phone Extension, Fax# and Email). Except for Password, the remaining fields which cannot be edited are periodically synchronized from the authentication server.



For more information on authentication data synchronization, see the `-authsynchinterval` broker argument in *Configuring External Authentication in the Implementation Guide*.

When you are using external authentication, if any of the following options are provided when importing or overwriting a user, they are ignored. A warning message displays for each option.

- Password
- RealName
- Phone#
- Phone Extension
- Fax#
- Email
- The `-cpw` option (change password at next logon)

### **Import User Mode**

When you are using internal authentication (CA Harvest SCM authentication), a user is created only if the User Name provided does not already exist in CA Harvest SCM.

When you are using external authentication, such as Microsoft Active Directory, a user is created only if the User Name provided does not exist in CA Harvest SCM and it exists in the authentication server. The input file format is:

```
UserName<tab>Password<tab>RealName<tab>  
Phone#<tab>Ext<tab>Fax#<tab>Email<tab>Note<tab>  
Usrg rp1<tab>Usrg rp2<tab>Usrg rp3<tab>...
```

### **Overwrite User Mode**

When you are using internal authentication (CA Harvest SCM authentication), the User Name can be changed only if the new User Name does not already exist in CA Harvest SCM.

When you are using external authentication such as Microsoft Active Directory, the User Name can be changed only if the new User Name does not already exist in CA Harvest SCM but it does exist on the authentication server.

When the `-nn` option is used, the input file format is:

```
CurrentUserName<tab>NewUserName<tab>Password<tab>RealName<tab>  
Phone#<tab>Ext<tab>Fax#<tab>Email<tab>Note<tab>  
Usrg rp1<tab>Usrg rp2<tab>Usrg rp3<tab>...
```

When the `-nn` option is *not* used, the input file format is:

```
UserName<tab>Password<tab>RealName<tab>
Phone#<tab>Ext<tab>Fax#<tab>Email<tab>Note<tab>
Usrgrp1<tab>Usrgrp2<tab>Usrgrp3<tab>...
```

## husrunlk Command-Unlock User

The `husrunlk` command unlocks a locked-out user (*LockedUserName* in the following syntax). If a user has consecutively used the wrong password and reached the limit of the `MaxFailAttemptBeforeLockout` setting, a locked out user can only log on after an administrator has unlocked the user.

The following considerations apply to authentication:

- When you are using internal authentication (CA Harvest SCM authentication), unlocking users is enabled.
- When you are using external authentication, such as Windows Active Directory, this functionality is disabled; unlocking users must be performed using methods provided by the authentication server. For example, if using Microsoft Active Directory, you can use Windows Domain User Manager to unlock a user.
- When you are using external authentication, unlocking users fails with an error message.

This command has the following format:

```
husrunlk {-b name} {-usr username} {-pw password} [-v] [-prompt] [-i inputfile.txt  
| -di inputfile.txt] [-eh filename] [-o filename | -oa filename] [-arg] [-wts] [-h]  
LockedUserName
```

**-b *name***

(Required) Specifies the CA Harvest SCM broker name.

**-usr *username***

(Required, unless `-prompt` is specified) Specifies the user name for login to the broker. If `-usr` is defined without `-pw`, the user is prompted for the password at runtime.

**-pw *password***

(Required, unless `-prompt` is specified) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the `-prompt` option.

**-eh *filename***

(Optional) Specifies the encrypted CA Harvest SCM username-password file name. The file name used with this option contains encrypted username-password credentials and must be generated with the svrenc utility.

**Note:** -eh overrides -usr and -pw options. -prompt overrides -eh, -usr and -pw options.

**Example: husrunlk command**

```
husrunlk -b brokername -usr username -pw password -o husrunlk.log BobSmith
```

**More information:**

[hppolset Command-Set Password Policy](#) (see page 99)

## svrenc Command-Encrypt User and Password Credentials to a File

The svrenc command is a user name and password encryption utility that stores encrypted credentials in a file, which can then be used by the:

- CA Harvest SCM Server to obtain the database user and password
- CA Harvest SCM Remote Agent (LDAP support)
- CA Harvest SCM command line utilities to obtain the CA Harvest SCM or remote computer user and password

This command has the following format:

```
svrenc {-s | -f filename} [-usr username] [-pw password] [-dir directory_name] [-o filename | -oa filename] [-arg] [-wts] [-h]
```

**-s**

(Required: -s and -f are mutually exclusive and one is required.) Specifies that the encrypted credentials are saved in a hidden file named: hsvr.dfo in <CA\_SCM\_HOME>; which is then utilized by the CA Harvest SCM Server processes when connecting to the Database Server.

**-f *filename***

(Required: -s and -f are mutually exclusive and one is required.) Specifies that an encryption file be created with the file name you provide.

If -f *filename* is specified, the encrypted credentials are saved in a hidden file named: *filename*, which can then be utilized by Remote Agent LDAP support or command-line utilities.

**-usr *username***

(Optional) Specifies the user name for login. If -usr is specified without -pw, the user is prompted for the password at runtime.

**-pw *password***

(Optional) Specifies the password for login to the broker. To hide the password from being exposed in plain text on the command line, use the -prompt option.

**-dir *directory\_name***

(Optional) Specifies the destination directory. If a directory is not specified, the encryption file is placed in CA\_SCM\_HOME. If a directory that does not exist is specified, the directory is created.

# Appendix A: Return Codes

---

**Note:** The instances of Linux in this section refer to both the Linux and zLinux operating environments.

All command line programs return the codes listed in Common Cases in this appendix. A few command line programs also return other codes in special cases.

This section contains the following topics:

[Return Codes and Log Files](#) (see page 173)

[Common Cases](#) (see page 174)

[Special Case - hchu](#) (see page 174)

[Special Case - hco](#) (see page 174)

[Special Case - hexecp](#) (see page 175)

[Return Code Examples](#) (see page 175)

## Return Codes and Log Files

CA Harvest SCM command line programs all return zero to indicate success and non-zero to indicate failure. In each case, additional information is logged in the command line program log file. You can use the -o option to specify a log file name. If you do not supply the -o option, the log file defaults to the current working directory and is named *progrname.log* (for example, hco.log, hcp.log, and so on).

## Common Cases

The majority of CA Harvest SCM command line programs return the same set of return codes:

- 0 Success
- 1 Command syntax is incorrect
- 2 Broker not connected
- 3 The command line program failed in some anticipated way
- 4 Unexpected error
- 5 Invalid login
- 6 Server or database down
- 7 Incorrect service pack level
- 8 Incompatible server version
- 9 Exposed password
- 10 Ambiguous arguments
- 11 Access denied
- 12 Prelink failed
- 13 Postlink failed

## Special Case - hchu

Password changes executed from the command line using hchu are disabled when external authentication is enabled. For example, the following command returns error code 94:

```
hchu -b brokername -sd-sun -usr username -pw password -ousr user4 -npw newpassword
```

## Special Case - hco

If no version was found for the file name or pattern you specified as hco command line arguments, hco displays an error to indicate it found no version for that file name or pattern. The hco command also returns error code 14 for this type of error.

Error code 14 is a low priority error code. If this error occurs with other types of errors on the same hco command, then hco will return the error code for the error with higher priority. For example, if a no version found error and an anticipated error (error code 3) occur together, then hco returns error code 3, which has a higher priority.

## Special Case - hexecp

Hexecp enables the invocation of remote programs. It can be run either in synchronous mode or asynchronous mode. The remote program may or may not return a return code. The return codes are modified as follows:

### **Asynchronous Mode**

0 indicates the remote program was found and invoked. 1, 3, and 4 are returned as in the common cases.

### **Synchronous Mode**

If the remote program is found and invoked then its return code is returned as the hexecp return code.

1, 3, and 4 are returned as in the common cases if the remote program is not invoked.

2 is returned if the remote program is invoked but does not return a code of its own.

## Return Code Examples

Distinct operating systems and various UNIX and Linux shells have different ways of distinguishing program return codes. The following examples are not meant to be robust or efficient.

## Windows

Windows batch files use the errorlevel. This system variable is used with an 'if' statement such that if the program return code is greater than or equal to that listed, then the condition is true. Larger values must be checked prior to smaller values.

```
<execute program>
if errorlevel 3 goto cmdline_error
if errorlevel 1 goto syntax_error
if errorlevel 0 goto success
;
:success
echo "Program completed successfully."
goto end
;
: cmdline_error
echo "Program failed."
goto end
;
: syntax_error
echo "Invalid command line syntax."
goto end
;
:end
echo "See log file for details."
```

## UNIX Bourne Shell

```
<execute program>
result=$?
case "$result"
in
    0) echo " Program completed successfully.;;"
    1) echo " Invalid command line syntax.;;"
    3) echo " Program failed.;;"
    4) echo " Program failed.;;"
esac

echo "See log file for details."
```



## Perl Script

The Perl script system return code must be shifted right eight bits to extract the actual program return value.

```
$sysarg = "<program command line>";  
$rc = system( $sysarg);  
  
if($rc > 0x80)  
{  
    $rc >>=8;  
}  
if ($rc == 0) { print " Program completed successfully.";}  
elsif ($rc == 1){print " Invalid command line syntax.";}  
else {print " Program failed.";}
```



# Index

---

## A

accessing reports • 16  
adding form attachments • 73  
approving packages • 18  
archiving data • 20  
authentication synchronization • 22

## C

changing item types • 37  
changing passwords • 39  
check-in process • 41  
check-out process • 47  
codes, return codes • 174  
command options • 9  
commands  
    caident • 15  
    haccess • 16  
    hap • 18  
    har • 20  
    hauthsync • 22  
    hcbl • 34  
    hccmrg • 36  
    hchgtype • 37  
        examples • 37  
        Global Options • 100  
        syntax • 37  
    hchu • 39  
    hci • 41  
    hcmpview • 46  
    hco • 47  
    hcp • 54  
    hcpj • 55  
    hcropmrg • 57  
    hcrtpath • 61  
    hdlp • 66  
    hdp • 67  
    hdv • 68  
    hexecp • 70  
    hexpenv • 72  
    hfatt • 73  
        syntax • 73  
    hformsync • 75  
    hft • 77  
    hgetusg • 78

himpenv • 79  
    example • 79  
    syntax • 79  
hlr • 80  
hlv • 82  
hmvpkg • 86  
    syntax • 86  
hpg • 92  
hpkgunlk • 94  
    command • 94  
    syntax • 94  
hpolset  
    access • 104  
hpp • 95  
hppolget • 97  
hpolset • 99  
    example • 106  
    user-level overrides • 103  
hrefresh • 107  
    syntax • 107  
hrepedit • 109  
    examples • 109  
    syntax • 109  
hrepmgr • 112  
hri • 125  
hrnitm • 129  
hrt • 132  
hsigget • 134  
    attributes • 134  
hsigset • 139  
    attributes • 139  
hsmtp • 141  
hspp • 142  
    examples • 142  
hsq1 • 145  
hsv • 147  
hsync • 151  
htakess • 156  
hudp • 161  
hup • 162  
husrmgr • 165  
    examples • 165  
husrunk • 170  
    syntax • 170  
move item • 84

---

- move path • 90
- remove item • 125
- remove path • 127
- rename path • 130
- svrenc • 171
- common cases, return codes • 174
- comparing view • 46
- concurrent merge process • 36
- configuring project baselines • 34
- copying projects • 55
- correlating packages to CMEW packages • 60
- creating empty repository paths • 109
- creating empty view paths • 61
- creating packages • 54
- cross project merge • 57

## D

- deleting packages • 66
- deleting snapshot views • 65
- deleting versions • 68
- demoting packages • 67

## E

- editing repositories • 109
- encrypting user/password • 171
- executing programs • 70
- exporting lifecycle definition • 72

## F

- footprinting • 77

## G

- generating password policy • 97
- generating reports • 145

## H

- haccess command • 16
- har command • 20
- hauthsync command • 22
- hcbl command • 34
- hccmrg command • 36
- hchgtype command • 37
  - examples • 37
  - syntax • 37
- hchu command • 39
- hci command • 41
- hcmpview command • 46

- hco command • 47
- hcp command • 54
- hcpj command • 55
- hcropmrg command • 57
- hcrllte command • 60
- hcrtpath command • 61
- hdelss command • 65
- hdlp command • 66
- hdp command • 67
- hdv command • 68
- hexecp command • 70
- hexpenv command • 72
- hfatt command • 73
  - syntax • 73
- hformsync command • 75
- hft command • 77
- hgetusg command • 78
- himpenv command • 79
  - example • 79
  - syntax • 79
- hlr command • 80
- hlv command • 82
- hmvpkg command • 86
  - syntax • 86
- hmvpth command • 90
- hpg command • 92
- hpolset command
  - access • 104
- hpp command • 95
- hpolget command • 97
- hpolset command • 99
  - example • 106
  - Global Options • 100
  - user-level overrides • 103
- hrefresh command • 107
  - syntax • 107
- hredit command • 109
  - examples • 109
  - syntax • 109
- hrepmgr
  - command • 112
  - create empty path
    - syntax • 124
  - create repository
    - syntax • 113
  - delete item or path
    - syntax • 122
  - delete repository
    - syntax • 114

---

- duplicate repository
  - syntax • 115
- load repository
  - syntax • 116
- move item or path
  - syntax • 121
- rename item or path
  - syntax • 122
- rename repository
  - syntax • 117
- update item or path property
  - syntax • 123
- update repository property
  - syntax • 118
- hri command • 125
- hrmvpth command • 127
- hrnitm command • 129
- hrnpth command • 130
- hrt command • 132
- hsigget command • 134
  - attributes • 134
- hsigset command • 139
  - attributes • 139
- hsmtp command • 141
- hspp command • 142
  - examples • 142
- hsql command • 145
- hsv command • 147
- hsync command • 151
- htakess command • 156
- hudp command • 161
- hup command • 162
- husrmgr command • 165
  - examples • 165
- husrunk command • 170
  - syntax • 170

## I

- importing lifecycle definitions • 79
- invoking external programs • 161

## L

- list version process • 82
- loading repositories • 80

## M

- maintaining package groups • 92
- maintaining user profiles • 165

- managing repositories • 112
- moving items • 84
- moving packages • 86
- moving paths • 90
- moving repository items • 109
- moving versions • 142

## N

- name values
  - specifying • 13

## O

- option arguments
  - specifying • 14
- options, command options • 9

## P

- password policy • 100
  - usage • 104
- password properties • 12
  - example • 12
- passwords
  - defining validation rules • 100
  - policy • 100
- promoting packages • 95
- properties, password • 12
  - example • 12

## Q

- querying signature file • 134

## R

- reference directories • 107
- refreshing directories • 107
- removing form attachments • 73
- removing items • 125
- removing paths • 127
- renaming items • 129
- renaming paths • 130
- renaming repositories • 109
- restoring data • 132
- retrieving form attachments • 73
- return codes
  - common cases • 174
  - examples • 175

---

## S

- selecting item versions • 147
- sending mail • 141
- setting password policy • 99
- specifying
  - name values • 13
  - option arguments • 14
- svrenc command • 171
- synchronizing directories • 151

## T

- taking snapshots • 156

## U

- unlocking packages • 94
- unlocking users • 170
- updating form templates • 75
- updating packages • 162
- updating signature file • 139
- user-defined process • 161