

Surround SCM CLI Reference Guide



August 2003

Copyright

© Copyright Seapine Software, Inc., 1996-2003. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the written permission of Seapine Software, Inc.

**Software License
Notice**

Your license agreement with Seapine Software, Inc., which is included with the product, specifies the permitted and prohibited uses of the product. Any unauthorized duplication or use of this software, in whole or in part, in print, or in any other storage and retrieval system is forbidden.

**Licenses and
Trademarks**

TestTrack, SoloBug, and SoloSubmit are registered trademarks of Seapine Software, Inc. Seapine, the Seapine logo, the Seapine License Server, QA Wizard, and Surround SCM are trademarks of Seapine Software, Inc. Windows is a registered trademark of Microsoft Corporation. Visual Basic, Visual C++, and Visual Studio .NET are registered trademarks of Microsoft Corporation. PVCS is a registered trademark of MERANT, Inc. Perforce is a registered trademark of Perforce Software, Inc. Palm OS is a registered trademark of Palm, Inc. Adobe Acrobat Reader is a registered trademark of Adobe Systems, Incorporated. CS-RCS is a registered trademark of ComponentSoftware, Inc. SourceOffSite Classic is a trademark of SourceGear Corporation. StarTeam is a registered trademark of Fox Data, Inc. ClearCase is a registered trademark of Rational Software Corporation. XML Spy is a registered trademark of Altova. Xerces and Xalan are registered trademarks of Apache Software Foundation. All third-party software and/or code is the property of their respective owners. The right to copyright and trademark this software has been granted. See your Seapine license agreement for additional information.

Information in this document is subject to change without notice and does not represent a commitment on the part of Seapine Software, Inc. Unless otherwise noted, all companies, products, street addresses, and persons contained herein are purely fictitious.

Mention of third party companies and products is for informational purposes only and does not constitute an endorsement. Seapine Software, Inc. assumes no responsibility with regard to the selection, performance, or use of these products. All understandings, agreements or warranties, if any, take place directly between the vendor and prospective users. agreements or warranties, if any, take place directly between the vendor and prospective users.

Seapine Software, Inc.

5390 Courseview Drive, Suite 115
Mason, OH 45040
513.754.1655
documentation@seapine.com

0340-2.0

Printed in the USA

Contents

Chapter 1	Understanding Surround SCM.....	1
	Surround SCM architecture.....	2
	What's new in Surround SCM 2.0?.....	2
	About the guide.....	2
	Documentation conventions.....	3
	Contacting Seapine support.....	3
 Chapter 2	 Setting up Surround SCM.....	 5
	Starting Surround SCM.....	6
	Configuring client options.....	6
	Setting a working directory.....	7
	Creating a mainline branch.....	7
	Adding files.....	8
 Chapter 3	 Surround SCM CLI Commands.....	 9
	About Surround SCM commands.....	10
	Common command options.....	10
	Branch commands.....	10
	addexistingmainline.....	11
	branchhistory.....	11
	branchproperty.....	11
	editshadow.....	12
	freeze.....	12
	lsbranch.....	13
	lsmainline.....	13
	lshadow.....	13
	mkbranch.....	14
	mkmainline.....	14
	mkshadow.....	15
	promote.....	15
	rebase.....	16
	renamebranch.....	16
	restorebranch.....	17
	rmbranch.....	17
	rmmainline.....	18
	rmshadow.....	18
	unfreeze.....	18
	upgrademainlinebranch.....	19
	File commands.....	19
	add.....	19
	batch.....	20
	breakshare.....	20

checkin.....	21
checkout	22
diff.....	23
get.....	23
history.....	24
label	25
ls.....	25
merge.....	26
promotefile.....	27
property	27
rebasefile.....	28
rename.....	29
reporthistory	29
restore.....	30
rm	30
rollback.....	31
share	32
uncheckout.....	32
voidcheckout.....	33
Repository commands.....	33
diff.....	33
history.....	34
label	34
merge.....	35
mkrepository	35
promotefile.....	36
property	36
rebasefile.....	37
rename.....	38
restore.....	38
rm	39
rmworkdir	39
securerepository.....	40
Security group commands	40
addgroup.....	40
editgroup.....	41
lsgroup.....	41
rmgroup	42
Server commands	42
editmailqueue.....	42
lsmailqueue	42
lsserverlog	42
rmserverlog.....	43
serveremailoption	44
serveroption.....	45
setclient	46
version.....	47
TestTrack Pro integration commands	47
fetchttddb.....	47

	lsdefect.....	48
	lsttdb.....	48
	rmtdb.....	48
	settdb.....	48
	Trigger commands.....	49
	addemailnotification.....	49
	addtrigger.....	50
	editemailnotification.....	52
	edittrigger.....	53
	lsemailnotification.....	54
	lstrigger.....	54
	rmemailnotification.....	55
	rmtrigger.....	55
	User commands.....	55
	adduser.....	55
	edituser.....	56
	lsuser.....	57
	passwd.....	57
	retrieveuser.....	58
	rmuser.....	58
	Working directory commands.....	58
	showworkdirs.....	58
	workdir.....	58
	workdirinfo.....	59
Chapter 4	CLI Quick Reference.....	61
	Quick reference commands.....	62
Chapter 5	Surround SCM Glossary.....	67
	Definitions.....	68
	Ancestor branch.....	68
	Baseline branch.....	68
	Branches.....	68
	Branch security.....	68
	Diff.....	68
	Email triggers.....	68
	File types.....	68
	Keywords.....	68
	Mainline branch.....	69
	Merge.....	69
	Pre-event triggers.....	69
	Post-event triggers.....	69
	Promote.....	69
	Rebase.....	69
	Recursive.....	69
	Repository.....	69
	Server security.....	70
	Shadow folder.....	70

Snapshot branch	70
Timestamp	70
Triggers	70
Working directory.....	70
Workspace branch	70
Index.....	71

Chapter 1

Understanding Surround SCM

You have an important job to do - deliver a quality product on time and within budget. And, whether you are developing, testing, fixing bugs, or managing the team, you do not have time to learn another complex application.

Surround SCM manages all of your team's digital assets, including source code, requirements documents, Web pages, image files, specifications, and more. With Surround SCM, your team can archive and version virtually any file and quickly locate and retrieve it at any time. Surround tracks who, when, and what changed for every revision, so you can always retrieve the best version of a file.

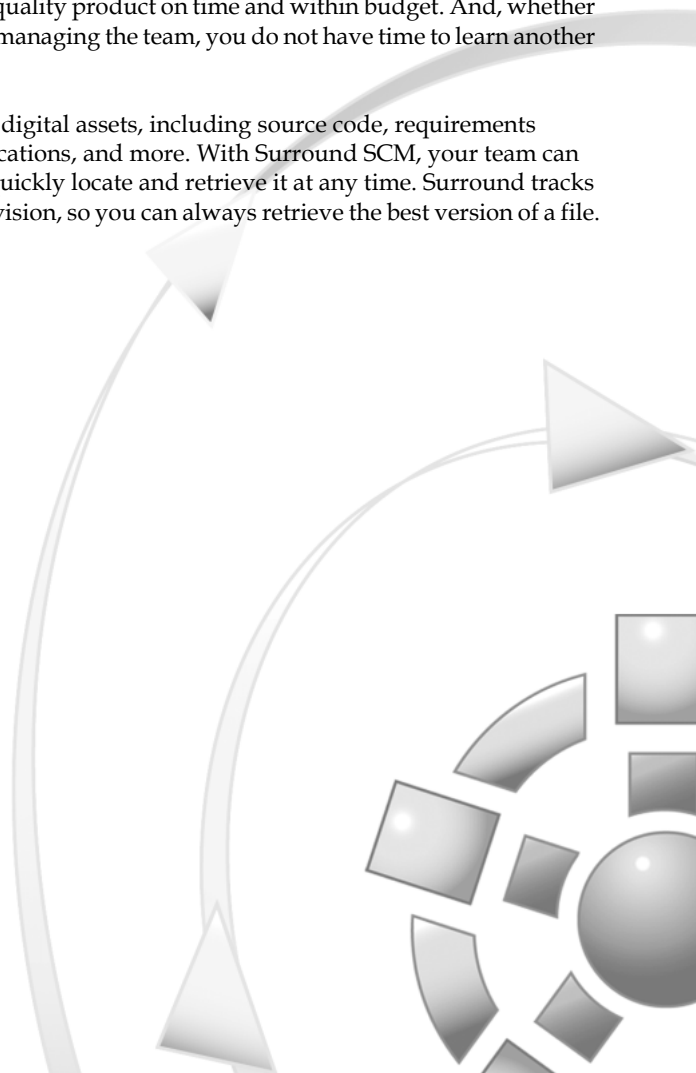
Surround SCM architecture, 2

What's new in Surround SCM 2.0?, 2

About the guide, 2

Documentation conventions, 3

Contacting Seapine support, 3



Surround SCM architecture

Surround SCM is a multi-threaded client/server cross-platform solution that easily scales to accommodate large product development teams. Command line and graphical user interface (GUI) clients that run on Windows, Linux, Solaris, and Mac OS X provide fast, reliable access to your source repository. MS SCC API support means Surround SCM works with a variety of popular Windows-based development tools, including Visual Studio.

The Surround SCM server is a required component. Based on the TCP/IP protocol, Surround's server easily supports geographically dispersed product development team. The Surround SCM server can be installed on any platform too - Windows, Mac, Linux, or UNIX.

Surround SCM's cross-platform flexibility lets you install the program to best meet your company's needs. Typically, the Surround SCM server is installed on a shared computer and client components are installed on each user's computer.

What's new in Surround SCM 2.0?

- The Seapine License Server - used to manage global licenses and users
- Email notifications
- Event triggers that fire an email or script pre- or post-event
- Ability to permanently delete files/repositories/branches
- Ability to rename files/repositories/branches
- User options for default check out values
- Integration with Macromedia Dreamweaver
- Integration with Borland JBuilder

About the guide

This guide provides step-by-step instructions for all the tasks you perform when working with Surround SCM. This guide is not necessarily meant to be read from start to finish. It includes information for users, and administrators, at a variety of levels. To get the most out of the documentation, start by reading the sections that are most relevant to your use of Surround. Admin-related tasks are clearly marked.

Documentation conventions

There are a few conventions used throughout the guide that are designed to be completely predictable – making it is easy to understand what you are reading and what you’re supposed to do.

Many of the commands are available from menus, toolbars, shortcut menus, and shortcut keys. Throughout this guide, most commands are explained using the menu commands. As you become more familiar with Surround SCM, you may find shortcut menus and shortcut keys more efficient to work with. When you are instructed to select a menu command, you will find the menu name, followed by an arrow. For example, to add a defect, choose **Activities > Get**. Indented text is also used to draw attention to notes, tips, examples, etc.

Contacting Seapine support

We offer technical support, 9 AM - 6 PM, EST, Monday through Friday.

Telephone: (513) 754-1655

Email: support@seapine.com

Web site: <http://www.seapine.com>

Documentation feedback

Seapine Software welcomes your feedback on the documentation. If you have comments or suggestions, please email: documentation@seapine.com, which is provided for documentation feedback only. You may not receive a reply to your email.

Chapter 2

Setting up Surround SCM

It only takes a few minutes for new and upgrade users to set up Surround SCM. After creating or upgrading mainline branches, your users will be working with source files in no time at all.

Starting Surround SCM, 6

Configuring client options, 6

Setting a working directory, 7

Creating a mainline branch, 7

Adding files, 8



Starting Surround SCM

Note: Before starting Surround SCM, make sure the Seapine license server and Surround SCM server are both running.

- 1 Change (*cd*) to the Surround SCM directory. Generally, *c:\program files\Seapine\Surround SCM*.

For example, *cd c:\Program Files\Seapine\Surround SCM*

You are now ready to start using the Surround SCM command line client. With few exceptions, the command line and graphical user interfaces provide the same functionality. See **Chapter 3, “Surround SCM CLI Commands,”** page 9 for detailed command information.

Configuring client options

- 1 At the command prompt, enter *sscm setclient*.

This command sets the client options and then displays the options.

Syntax

sscm setclient [-dDiffUtility] [-h] [-mMergeUtility] [-w | -w-] [-c | -c-] [-i | -i-] [-jPath] [-gPath] [-n | -n-] [-r | -r-] [-yUsername:Password] [-zSCMServerAddr:PortNum]

<i>-d</i>	Enter the diff utility to be used by command diff. Must include the full path of the diff utility, %1 for the position of first file and %2 for the position of the second file, and/or other options for the utility. The default utility is used if an option is not specified.
<i>-m</i>	Enter the 3rd party 3-way merge utility provided by user. It must include the full path of the merge utility package, %1 for the position of the common version file, %2 for the position of the newer version file, %3 for the position of the local changed file, and/or other options of the utility. The default utility is used if an option is not specified.
<i>-w</i>	Specifies files are in Windows text format. <i>-w</i> for Windows and <i>-w-</i> for Unix.
<i>-c</i>	Set data compression during file transfer on with <i>-c</i> or off with <i>-c-</i> .
<i>-i</i>	Use the integrated Guiffy diff/merge utility with <i>-i</i> or use selected diff/merge utilities with <i>-i-</i> .
<i>-j</i>	Set the Java runtime path for use with the Guiffy diff/merge utility.
<i>-g</i>	Set the path for use with the Guiffy diff/merge utility.
<i>-n</i>	Set option to update version even if no change is made on with <i>-n</i> or off with <i>-n-</i> .
<i>-r</i>	Set local file removal on check in on with <i>-r</i> or off with <i>-r-</i> .

Setting a working directory

You must set a working directory before you can work with source files (get, check out, check in, etc.). The working directory is generally mapped to a directory on your local drive. You can also choose a different location for the working directory, such as a network drive.

- 1 At the command prompt, enter `sscm workdir`.

Syntax

`sscm workdir Directory Repository [-bBranch] [-h] [-r] [-yUsername:Password] [-zSCMServerAddr:PortNum]`

<i>Directory</i>	Enter the full directory path.
<i>Repository</i>	Enter the full repository path.
<i>-b</i>	Enter the branch name. No branch option means main branch.
<i>-r</i>	Recursively set the working directories for the subrepositories in the branch.

Creating a mainline branch

A mainline branch is the highest-level, central branch that contains all source files, labels, other branches, and repositories. All files saved to a specific Surround SCM server are stored in a corresponding branch. At least one mainline branch must be created before source files can be added to Surround SCM.

You cannot search, branch, promote, or rebase across mainline branches. Take the time to decide on your company's use of mainline branches. Some companies may need only one mainline branch while other companies may have multiple mainline branches.

- 1 At the command prompt, enter `sscm mkmainline`.

Syntax

`sscm mkmainline MainLineName [-ccComment | -cpCommentPath | -c-] [-h] [-lMainLineLoc]
[-yUsername:Password] [-zSCMServerAddr:PortNum]`

- c Enter a comment or path to the comment text file and the prompt will not appear. No -c means comment input prompt will appear. -c- means no comment needed.

CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.

- l Enter the mainline branch location on the server. The mainline branch and relative files are stored in this location. The default name is the same as the mainline branch name.

Adding files

You can add text, binary, and Mac binary files to Surround SCM. Surround SCM supports scripts, test drivers, documentation, notes, software files, HTML, and most any other type of file. After files are added, all changes are controlled by Surround SCM. Additional files or folders can be added at any time. Surround SCM lets you add single files, multiple files, or all files in a directory.

Note: The main Surround SCM command is *sscm*. The case sensitivity of this command is platform-dependent. Other commands and options are case insensitive. For example, *checkin* and *-p* can also be typed as *CHECKIN* or *-P*.

- 1 At the command prompt, enter *sscm add*.

Syntax

```
sscm add Item1 [Item2 ...] [-bBranch] [-ccComment | -cpCommentPath | -c-] [-g | -g-][[-h] [-k | -ke]
[-pRepository] [-q] [-r] [-tFileType] [-u] [-w] [-yusername1:password1] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the full path of a local file or directory or the name of a file or subdirectory in the current directory. * for all files and subdirectories in the current directory and for the current directory.
<i>-b</i>	Enter the branch name. The default branch is set in the working directory.
<i>-c</i>	Enter a comment or the comment text file path. Enter <i>-c</i> to prompt for comment input. Enter <i>-c-</i> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<i>-g</i>	Get file after add for <i>-g</i> and do not get for <i>-g-</i> . Default is <i>-g</i> .
<i>-k</i>	Lock the file immediately after adding it. <i>-ke</i> for exclusive lock. The local file is still writable.
<i>-p</i>	Enter the full repository path.
<i>-q</i>	Quiet mode. Do not list repository and local full path of the Surround SCM server.
<i>-r</i>	Recursively add files and repositories.
<i>-t</i>	Enter the file type. Can be <i>auto_detect</i> , <i>text</i> , <i>binary</i> and <i>mac_binary</i> . On Mac platform, only the files created by Mac editors can be automatically detected with their types. Enter the file type using this option if a file is migrated from another platform.
<i>-u</i>	Undelete the file and then check in the new one if it was deleted.
<i>-w</i>	Make file writable after add.

Chapter 3

Surround SCM CLI Commands

This section provides an explanation of Surround SCM CLI commands. It includes the syntax for each command and any command options.

About Surround SCM commands, 10

Common command options, 10

Branch commands, 10

File commands, 19

Repository commands, 33

Security group commands, 40

Server commands, 42

TestTrack Pro integration commands, 47

Trigger commands, 49

User commands, 55

Working directory commands, 58



About Surround SCM commands

The main Surround SCM command is `sscm`. The case sensitivity of this command is platform-dependent. Other commands and options are case insensitive. For example, `checkin` and `-p` can also be typed as `CHECKIN` or `-P`.

In the following commands, except `passwd`, the Surround SCM username and password and the Surround SCM server host computer address and port number can be set in the client. These values become the default values. You can use the default values or specify other values. The default branch name and repository name are set in the working directory.

For all commands, if an argument contains a space, quote the argument. This tells the command it is one argument. For example: `sscm add "my files"`.

Common command options

The following options are used with most commands.

<code>-h</code>	Request online help for the command.
<code>-y</code>	If you do not want to use the default value, enter a Surround SCM username and password.
<code>-z</code>	If you do not want to use the default value, enter the Surround SCM server host computer address and port number.

Branch commands

A branch is a separate line of development that uses an existing repository, and the files in that repository, as a starting point. When a repository is branched, the files in both branches are initially identical. As file contents change, the branched files become dissimilar. Use branches when you need to make changes to source files without affecting files in the existing repository. When you branch a repository, changes can be promoted to, or rebased from, associated branches.

A **mainline branch** is the highest-level branch that contains all source files, labels, other branches, and repositories. All files saved to a specific Surround SCM server are stored in a corresponding repository.

A **baseline branch** is a public branch. Allowing checkouts on a baseline branch lets all users directly check out and make changes to the baseline code. Changes made to the baseline branch affect everyone who accesses that branch.

A **workspace branch** is a private branch that is used to track and isolate changes a user makes. The branch a workspace is created from its parent branch. Changes can be promoted to or rebased from the parent branch. Other users are not affected by any changes that are made in a private, workspace branch.

A **snapshot branch** is a static branch of a baseline or workspace branch that generally corresponds to a project milestone, such as a QA build or final release build. Most Surround SCM commands are disabled in snapshot branches.

addexistingmainline

When you remove a mainline branch, it is not deleted. You can add an existing mainline branch to make it active again.

Shortcut command: **aeml**

Syntax

sscm addexistingmainline *MainlineBranchLocation* [-h] [-yUsername:Password]
[-zSCMServerAddr:PortNum]

branchhistory

View the history events for a branch that has been removed but not destroyed.

Shortcut command: **bh**

Syntax

sscm branchhistory [-bBranch [-pBaseRepository]] [-dDateFrom:DateTo] [-uUsername] [-h]
[-yUsername:Password] [-zSCMServerAddr:PortNum]

-b	Enter the branch name. The default branch is set in the working directory.
-d	Display the branch history between the two specified dates. Date format: <i>yyyymmdd</i>
-p	Enter the full base repository path.
-u	Display changes made by a specific user.

branchproperty

Set, and display, branch properties. You can apply repository level security to a branch, overriding the server level security. When you configure branch security, you are setting repository level security that only affects one branch. For example, you can restrict access to a branch of released code.

Shortcut command: **bp**

Syntax

sscm branchproperty [-bBranch] [-h] [-i | -o | -si | -so | -sd] [-pMainlineBranch] [-yUsername:Password]
[-zSCMServerAddr:PortNum]

-b	Enter the branch name. If a branch is not specified, the default branch is used.
-i	Inherit parent branch's security.
-o	Use security that applies to all branches.
-p	Enter the mainline branch name.

-s	Branch uses its own security. Enter where the initial security settings should come from. -si to copy the parent branch's security settings. -so to copy the security that applies to all branches. -sd to copy the server level default security settings.
----	---

editshadow

Edit or force shadow folders to update.

Shortcut command: **es**

Syntax

sscm editshadow [[*RepositoryPath* [-b*Branch*]] | -l*Path*] [-f] [-r | -r-] [-w | -w-] [-tcurrent | -tmodify | -tcheckin] [-h] [-y*Username:Password*] [-zSCMServerAddr:portN]

<i>RepositoryPath</i>	The repository path of the shadow folder to be updated.
-b	The branch of the shadow folder to be updated.
-l	The server directory of the shadow folder to be updated.
-f	Force an update of the shadow folder contents.
-r	Change the recursive option. -r for recursive and -r- for non-recursive.
-w	Change the text file format. -w for Windows format and -w- for Unix format.
-t	Change the timestamp to use for shadowed files.

freeze

Freezing a branch prevents any code changes being made to files in the branch. When a branch is frozen, it is locked and no changes can be made to it.

Syntax

sscm freeze *Branch* [-h] [-p*Mainline*] [-y*Username:Password*] [-zSCMServerAddr:portNum]

<i>Branch</i>	Enter the branch name.
-p	Enter the mainline branch name. The default is defined in the working directory.

lsbranch

List all the branches in a repository and their corresponding parent branches.

Shortcut command: **lb**

Syntax

`sscm lsbranch [-fFile] [-h] [-pRepository] [-d] [-w] [-yUsername:Password] [-zSCMServerAddr:PortNum]`

<code>-d</code>	List deleted items.
<code>-f</code>	Enter a file name. Only the branches including the file will be listed.
<code>-p</code>	Enter the full repository path.
<code>-w</code>	List other users' workspace branches.

lsmainline

List all mainline branches.

Shortcut command: **lml**

Syntax

`sscm lsmainline [-h] [-yUsername:Password] [-zSCMServerAddr:PortNum]`

lshadow

List shadow folders.

Shortcut command: **lss**

Syntax

`sscm lshadow [[-pRepository [-bBranch]] | -a-] [-h] [-yUsername:Password] [-zSCMServerAddr:portN]`

<code>-p</code>	The repository path of the shadow folder to list.
<code>-b</code>	The branch of the shadow folder to list.
<code>-a</code>	List all shadow folders. Cannot be used with the <code>-p</code> or <code>-b</code> options.

mkbranch

Create a new branch.

Shortcut command: **mb**

Syntax

```
sscm mkbranch Branch Repo [-bParentBranch][-ccComment | -cpCommentPath | -c-] [-h] [-lLabel]
[-sworkspace | -sbaseline | -ssnapshot] [-tTimeStamp] [-yUsername:Password]
[-zSCMServerAddr:PortNum]
```

<i>Branch</i>	Enter the name of the branch you want to create.
<i>Repo</i>	Enter the full repository path.
<i>-b</i>	Enter the parent branch you want to create the new, child branch from. If this option is not specified, the mainline branch is used.
<i>-l</i>	Enter the label. This option specifies which parent branch file versions are copied into the child branch.
<i>-s</i>	Enter the type of branch you want to create: <i>workspace</i> (default), <i>baseline</i> , or <i>snapshot</i> .
<i>-t</i>	Enter the timestamp (local time). This option specifies which parent branch file versions are copied into the child branch. Date/time format: <i>yyyymmddhh:mm:ss</i> You cannot use both the <i>-t</i> and <i>-l</i> options. If neither option is specified, the newest file versions are used.

mkmainline

Create a mainline branch.

Shortcut command: **mml**

Syntax

```
sscm mkmainline MainLineName [-ccComment | -cpCommentPath | -c-] [-h] [-lMainLineLoc]
[-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>MainLineName</i>	Enter the new mainline branch name.
<i>-l</i>	Enter the mainline branch location on the Surround SCM server. The mainline branch and relative files are stored in this location.

mkshadow

Create a shadow folder.

Shortcut command: **mks**

Syntax

```
sscm mkshadow Path [-pRepository [-bBranch]] [-r | -r-] [-w | -w-] [-tcurrent | -tmodify | -tcheckin] [-h]
[-yUsername:Password] [-zSCMServerAddr:portN]
```

<i>Path</i>	The directory on the server in which to shadow folders. This can be a relative or absolute path. An empty directory must be manually created on the server before creating the shadow folder.
<i>-p</i>	The full path of the repository to shadow. Default repository is set in the working directory.
<i>-b</i>	The branch the shadow folder is being created for. Default branch is set in the working directory.
<i>-r</i>	Recursively shadow all subrepositories. <i>-r</i> for recursive and <i>-r-</i> for non-recursive. Default is non-recursive.
<i>-w</i>	Enter the text file format. <i>-w</i> for Windows format and <i>-w-</i> for Unix format. Default is the server's text file format.
<i>-t</i>	Specifies the timestamp to use for shadowed files. <i>-tcurrent</i> uses the current time. <i>-tmodify</i> uses the modification time. <i>-tcheckin</i> uses the check in time. The default is the current time.

promote

Update parent branch or grandparent branches with the changes in a child branch.

Syntax

```
sscm promote ChildBranch [-ccComment | -cpCommentPath | -c-] [-h] [-pRepository] [-s] [-uUpperBranch]
[-v] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>ChildBranch</i>	Enter the child branch name. To specify the default branch enter @.
<i>-p</i>	Enter the full repository path.
<i>-s</i>	Skip the server auto-merge.
<i>-u</i>	Enter the ancestor branch to be updated. This branch can be the parent branch, a grandparent branch, etc. If this option is not specified, the parent branch is used.
<i>-v</i>	Preview the result of promoting the changes files.

rebase

Update a child branch with the changes in the parent branch.

Syntax

sscm rebase *ChildBranch* [-b*SnapshotBranch*] [-cc*Comment* | -cp*CommentPath* | -c-] [-h] [-l*Label*] [-p*Repository*] [-s] [-t*Timestamp*] [-v] [-y*Username:Password*] [-zSCMServerAddr:PortNum]

<i>ChildBranch</i>	Enter the child branch name. To specify the default branch enter @ .
-b	Enter a snapshot branch name. This determines which file versions are copied into the child branch. Note: The -b, -l, and -t options are mutually exclusive. If one of these options is not specified, the current parent branch version is copied.
-l	Enter a label to specify which parent branch file versions are copied to the child branch.
-p	Enter the full repository path.
-s	Skip the server auto-merge.
-t	Enter the timestamp (local time). This determines which file versions are copied into the child branch. Date/time format: <i>yyyymmddhh:mm:ss</i>
-v	Preview the result of rebasing the changed files.

renamebranch

Rename an existing branch.

Syntax

sscm renamebranch *CurrentName* *NewName* [-p*BaseRepository*][-cc*Comment* | -cp*CommentPath* | -c-] [-h] [-y*Username:Password*] [-zSCMServerAddr:Port]

<i>CurrentName</i>	Enter the existing branch name.
<i>NewName</i>	Enter the new branch name.
-b	Enter the name of the branch.
-p	Enter the name of the mainline branch which the repository you are renaming is stored in.

restorebranch

Depending on your security options, you can restore removed baseline, snapshot, and workspace branches. In addition, you can also restore other users' workspace branches.

To restore a snapshot branch, you must have security to create snapshot branches. To restore baseline branches, you must have security to create baseline branches. To restore your own workspace branches, you must have security to use workspace branches. To restore other users' workspace branches, you must have permission to destroy other users' workspace branches.

Syntax

```
sscm restorebranch BranchName [-pBaseRepository] [-uUserName | -iUserID] [-ccComment | -cpCommentPath] [-c-] [-h] [-yUsername:Password] [-zSCMServerAddr:Port]
```

<i>BranchName</i>	Enter the name of the branch you want to restore.
<i>-b</i>	Enter the name of the branch.
<i>-i</i>	Enter the user id. If the branch is another user's workspace branch, only one of the <i>-u</i> or <i>-i</i> arguments is used, otherwise neither is used.
<i>-p</i>	Enter the mainline branch name.
<i>-u</i>	Enter the user name. If the branch is another user's workspace branch, only one of the <i>-u</i> or <i>-i</i> arguments is used, otherwise neither is used.

rmbranch

Remove an existing branch.

Shortcut command: **rb**

Syntax

```
sscm rmbranch BranchName [-h] [-ccComment | -cpCommentPath] [-c-] [-pRepository] [-d] [-uUserName | -iUserID] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>BranchName</i>	Enter the name of the branch you want to remove.
<i>-d</i>	Permanently destroy the branch.
<i>-i</i>	Enter the user id. If the branch is another user's workspace branch, only one of the <i>-u</i> or <i>-i</i> arguments is used, otherwise neither is used.
<i>-p</i>	Enter the full repository path.
<i>-u</i>	Enter the user name. If the branch is another user's workspace branch, only one of the <i>-u</i> or <i>-i</i> arguments is used, otherwise neither is used.

rmmainline

Remove a mainline branch.

Shortcut command: **rml**

Syntax

sscm rmmainline *MainLineName* [-ccComment | -cpCommentPath | -c-] [-h] [-yUsername:Password]
[-zSCMServerAddr:PortNum]

<i>MainLineName</i>	Enter the name of the mainline branch.
---------------------	--

rmshadow

Remove a shadow folder.

Shortcut command: **rms**

Syntax

sscm rmshadow [*RepositoryPath* [-bBranch] [-a-]] | -lPath [-h] [-yUsername:Password]
[-zSCMServerAddr:portN]

<i>RepositoryPath</i>	The repository path of the shadow folder to be removed.
-b	The branch of the shadow folder to be removed.
-l	The server directory of the shadow folder to be removed.
-a	Remove all shadow folders for the specified branch and repository. Cannot be used with the -l option.

unfreeze

Unlock a frozen branch.

Shortcut command: **uf**

Syntax

sscm unfreeze *Branch* [-h] [-pMainline] [-yUsername:Password] [-zSCMServerAddr:portNum]

<i>Branch</i>	Enter the branch name.
-p	Enter the mainline branch name. The default is defined in the working directory.

upgrademainlinebranch

The Surround SCM database format may change between major, and some minor, releases. Database changes may be necessary to support new features and functionality. If the database format changes, you need to upgrade mainline branches to the new format. Before upgrading, make a backup copy of the mainline branch.

Shortcut command: **uml**

Syntax

sscm upgrademainlinebranch *MainlineBranch* [-d] [-h] [-yUsername:Password]
[-zSCMServerAddr:PortNum]

-d	Enter the name of the directory that contains the database to be upgraded. If a directory is not specified, the name of an outdated mainline branch is used.
----	--

File commands

add

When you add files, Surround SCM checks in the files. When you add a directory, a repository with the same name as the directory is created. Files are added to the corresponding repository. Surround SCM optionally repeats the process recursively for all subdirectories.

Syntax

sscm add *Item1* [*Item2 ...*] [-bBranch] [-ccComment | -cpCommentPath | -c-] [-g | -g-][-h] [-k | -ke]
[-pRepository] [-q] [-r] [-tFileType] [-u] [-w] [-yusername1:password1] [-zSCMServerAddr:PortNum]

<i>Item</i>	Enter the full path of a local file or directory or the name of a file or subdirectory in the current directory. To specify all files and subdirectories in the current directory enter <i>*</i> .
-b	Enter the branch name. The default branch is set in the working directory.
-c	Enter a comment or the comment text file path. Enter <i>-c</i> to prompt for comment input. Enter <i>-c-</i> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
-g	Get file after add. If you do not want to get the file, enter <i>-g-</i> .
-k	Lock the file immediately after adding it. To lock the file exclusively, enter <i>-ke</i> .
-p	Enter the full repository path.
-q	Quiet mode. Do not list repository and local full path of the Surround SCM server.
-r	Recursively add files and repositories.

<code>-t</code>	Enter the file type. Can be <i>auto_detect</i> , <i>text</i> , <i>binary</i> and <i>mac_binary</i> . On Mac platform, only the files created by Mac editors can be automatically detected. Enter the file type using this option if a file is migrated from another platform.
<code>-u</code>	Undelete the file and then check in the new one, if it was deleted.
<code>-w</code>	Make file writable after add.

batch

Processes the batch commands found in the input file. Each line in the input file should contain a single Surround SCM command including proper command line options. Main command `sscm`, Surround SCM server address, port number, username and password are not required for each command line.

Syntax

```
sscm batch InputFile [-h] [-oOutputFile][-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<code>-o</code>	Enter a text file to direct all standard output. When executing commands from the input file, all output will be written to this file rather than being displayed on the screen.
-----------------	--

breakshare

When files are unshared, a copy of all archive information is made to the shared directory. The link is broken after the files are successfully copied. If a base file is shared more than once, the shared status only changes for the selected files or repository.

Shortcut command: **bs**

Syntax

```
sscm breakshare [File1 File2 ...] [-bBranch] [-ccComment | -cpCommentPath | -c-] [-h]
[-pRepository] [-r] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>File</i>	Name of the file you want to break the share for. Can be / or empty, which means the repository specified by the -p option or the default repository.
<code>-b</code>	Enter the branch you want to break shares for. The default branch is set in the working directory.
<code>-p</code>	Enter the full path of the repository.
<code>-r</code>	Recursively break all files and sub-repositories.

checkin

Check in updates Surround SCM files with changes, removes the lock on the files, and makes changes available to other users.

Shortcut command: **ci**

Syntax

```
sscm checkin Item1 [Item2 ...] [-bBranch] [-ccComment | -cpCommentPath | -c-] [-f] [-g | -g-] [-h] [-k] [-lLabel]
[-pRepository] [-q] [-r] [[[sTTConfigName] -iTTUsername:TTPassword] -aDefectNum1[:DefectNum2:...]]
[-w] [-u | -u-] [-d | -d-] [-ySCMUsername:SCMPasssword1] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<i>-a</i>	Enter the TestTrack Pro defect number(s) for attachment.
<i>-b</i>	Enter the branch name to check in the changes to. The default branch is set in the working directory.
<i>-c</i>	Enter a comment or the comment text file path. Enter <i>-c</i> to prompt for comment input. Enter <i>-c-</i> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<i>-f</i>	Force check in without merge. This option ignores code changes checked in after the user's last checkout or merge.
<i>-g</i>	Get file after check in. If you do not want to get the file, enter <i>-g-</i> .
<i>-i</i>	Enter the TestTrack Pro username and password.
<i>-k</i>	Keep the lock after check in.
<i>-l</i>	Enter a label for the check in code.
<i>-p</i>	Enter the full repository path.
<i>-q</i>	Quiet mode. Do not list repository and local full path of the Surround SCM server.
<i>-r</i>	Recursively check in all files and sub-repositories.
<i>-s</i>	Enter the TestTrack Pro server configuration name.
<i>-w</i>	Make file writable after check in.
<i>-u</i>	Allow file check in even if no change is made.
<i>-d</i>	Remove local file after check in.

checkout

Check out files when you need to make changes. You can check out single files, multiple files, or a repository. Surround SCM creates a read-write copy of the file in the working directory.

Shortcut command: **co**

Syntax

```
sscm checkout Item1 [Item2...] [-bBranch] [-ccComment | -cpCommentPath | -c-] [-e] [-f] [-h] [-pRepository]
[-q] [-r] [-tcurrent | -tmodify | -tcheckin] [-vVersion] [-wprompt | -wreplace | -wskip]
[-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<i>-b</i>	Enter the name of branch to check out files from. The default branch is set in the working directory.
<i>-c</i>	Enter a comment or the comment text file path. Enter <i>-c</i> to prompt for comment input. Enter <i>-c-</i> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<i>-e</i>	Exclusively lock the files.
<i>-f</i>	Force file retrieval from server regardless of the local copy status.
<i>-p</i>	Enter the full repository path.
<i>-q</i>	Quiet mode. Do not list repository and local full path of the Surround SCM server.
<i>-r</i>	Recursively get files and sub-repositories.
<i>-t</i>	Set the local file's date/time: <i>current time</i> (default), <i>modification time</i> , <i>check in time</i> .
<i>-v</i>	Enter which file version to check out.
<i>-w</i>	Enter how to handle a local writable file: <i>prompt</i> (default), <i>replace</i> , <i>skip</i>

diff

You can diff repositories to quickly see if there are differences between the file in your working directory and a version of the file or between two versions of the file.

Syntax

```
sscm diff File [-bBranch1[:Branch2]] [-h] [-pRepository] [-uDiffUtility] [-vVersion1[:version2]]
[-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>File</i>	Enter the file name.
<i>-b</i>	Enter the branch name. No branch specified means the default branch set in the working directory will be used. You can compare files between the working directory and a branch, file versions within the same branch, or files between two branches.
<i>-p</i>	Enter the full repository path.
<i>-u</i>	Enter the diff utility you want to use. It must include the full path of the diff utility package, %1 for the position of the first file and %2 for the position of the second file, and/or other options of the utility. The default diff of the SCM product will not be used. The third party diff utility can be set as the default utility using the <i>setclient</i> command.
<i>-v</i>	Display the difference between the two versions.

get

Get files when you want to view a file but do not need to make any changes. You can get a single file, multiple files, or a repository. A read-only copy of the file is created in the specified directory.

Syntax

```
sscm get Item1 [Item2...] [-bBranch] [-dLocalDir] [-e] [-f] [-h] [-pRepository] [-lLabel] [-q] [-r]
[-tcurrent | -tmodify | -tcheckin] [-vVersion] [-wprompt | -wreplace | -wskip] [-yUsername:Password]
[-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<i>-b</i>	Enter the branch name. The default branch is set in the working directory.
<i>-d</i>	Enter the local directory you want to get the files to. If <i>Item</i> is a repository, a subdirectory, with the same name as the repository, is created and files are copied to it. If <i>Item</i> is specified as /, files are copied to the local directory. If an option is not specified, files are copied to the working directory.
<i>-e</i>	Make local file editable or writable. Default is read-only.
<i>-f</i>	Force file retrieval from server regardless of the local copy status.

<code>-p</code>	Enter the full repository path.
<code>-l</code>	Enter a label to search for when getting a file.
<code>-q</code>	Quiet mode. Do not list repository and local full path of files.
<code>-r</code>	Recursively get files and sub-repositories.
<code>-t</code>	Set the local file's date/time: <i>current time</i> (default), <i>modification time</i> , <i>check in time</i>
<code>-v</code>	Enter the file version to get.
<code>-w</code>	Enter how to handle a local writable file: <i>prompt</i> (default), <i>replace</i> , <i>skip</i>

history

Show the history of a file or a repository to view date, user, action performed and file version number.

Syntax

```
sscm history Item [-aAction] [-bBranch] [-dDateFrom:DateTo] [-h] [-pRepository] [-uUsername]
[-vVersionFrom:VersionTo] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<code>-a</code>	Display the file history by only listing the specified action. Actions include: AddToRepository, AddToBranch, AddFromBranch, CheckIn, Rebase, RebaseWithMerge, Promote, Label, AttachToDefect, Delete, Undelete.
<code>-b</code>	Enter the branch name. The default branch is set in the working directory.
<code>-d</code>	Display the file history between the two specified dates. Date format: <i>yyyymmdd</i>
<code>-p</code>	Enter the full repository path.
<code>-u</code>	Display changes made by a specific user.
<code>-v</code>	Display changes between the two versions.

label

Labels provide a way to mark a specific version of a file or repository. You can create labels for single files, multiple files, or all files in a repository. When you create a label, a new entry is created in the history. The file, and the version number, do not change.

Syntax

```
sscm label Item1 [Item2 ...] [-bBranch] [-ccComment | -cpCommentPath | -c-] [-h] [-lLabel] [-o] [-pRepository]
[-r] [-vVersion] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<i>-b</i>	Enter the branch name. The default branch is set in the working directory.
<i>-c</i>	Enter a comment or the comment text file path. Enter <i>-c</i> to prompt for comment input. Enter <i>-c-</i> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<i>-l</i>	Enter the new label. To prompt for input, do not enter an <i>-l</i> option.
<i>-o</i>	Overwrite the existing label.
<i>-p</i>	Enter the full parent repository path.
<i>-r</i>	Recursively label all files.
<i>-v</i>	Enter the version number of a file.

ls

Displays repository or file information. If the argument is a repository name, a list of all files and subrepositories is displayed. If the argument is not a valid repository, it is assumed to be a regular expression used to search file names. File information includes local file status, type, and modification date. Local file status includes old (local file version is older than server version), not exist (local file does not exist), changed (local file is modified comparing the checked out or gotten version), and no change.

Syntax

```
sscm ls [Item] [-bBranch] [-h] [-pRepository] [-r] [-eEventText] [-aEventType] [-m[F:E:FE]MatchCase]
[-x[F:E:FE]ParseAsRegularExp] [-uCheckoutUser] [-w] [-yUsername:Password]
[-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the file or repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<i>-b</i>	Enter the branch name. The default branch is set in the working directory.

<code>-p</code>	Enter the full repository path.
<code>-r</code>	Recursively display all files in a repository.
<code>-e</code>	Enter the event search text.
<code>-a</code>	Enter a specific event to search for.
<code>-m</code>	Specifies a case sensitive search. Enter <code>-F</code> to perform a case sensitive search on a file name. Enter <code>-E</code> to perform a case sensitive search on an event. Enter <code>-FE</code> to perform a case sensitive search on file name and event text.
<code>-x</code>	Specifies a regular expression search. Enter <code>-F</code> to interpret the filename as a regex. Enter <code>-E</code> to interpret the event text as a regex. Enter <code>-FE</code> to interpret both the filename and event text as regular expressions.
<code>-u</code>	Display the files checked out by the specified user. Enter <code>*</code> to display files checked out by all users.
<code>-w</code>	Display the working directory for the current user.

merge

You can easily merge changes between server files and files in your working directory.

Syntax

```
sscm merge File [-bBranch] [-h] [-pRepository] [-uMergeUtility] [-vVersion] [-yUsername:Password]
[-zSCMServerAddr:PortNum]
```

<i>File</i>	Enter the file name.
<code>-b</code>	Enter the branch name. If a branch is not specified, the default branch is used.
<code>-p</code>	Enter the full repository path.
<code>-u</code>	Enter the merge utility you want to use. It must include the full path of the merge utility package, %1 for the position of the common version file, %2 for the position of the newer version file, %3 for the position of the local changed file, and/or other options of the utility. The default merge tool of the SCM product will not be used. The third party merge utility can be set as the default utility using the <code>setclient</code> command.
<code>-v</code>	Merge from the specified version.

promotefile

Update a file or repository in an ancestor branch with file or repository changes in a child branch. You can promote all files in a repository without affecting other items in the branch.

Shortcut command: **pf**

Syntax

```
sscm promotefile Item1 [Item2 ...] [-bBranch] [-ccComment | -cpCommentPath | -c-] [-h] [-pRepository] [-r] [-s]
[-uUpperBranch] [-v] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the file or repository name. Can be / or empty, which means the repository specified by the -p option or the default repository.
<i>-b</i>	Enter the child branch name you want to promote changes from. The default branch is set in the working directory.
<i>-c</i>	Enter a comment or the comment text file path. Enter -c to prompt for comment input. Enter -c- if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<i>-p</i>	Enter the full repository path.
<i>-r</i>	Recursively promote all files and sub-repositories.
<i>-s</i>	Skip the server auto-merge.
<i>-u</i>	Enter the ancestor branch to be updated. This branch can be the parent branch, a grandparent branch, etc. If this option is not specified, the parent branch is used.
<i>-v</i>	Preview the results of promoting the changed files.

property

Set the properties of a file or a repository and then display the properties. This command can be used to set up the default TestTrack database configuration for a repository.

Syntax

```
sscm property Item [-bBranch] [-d] [-kon | -koff | -kdefault] [-h] [-i] : [-o | -o- -sTTDbConfigName -adisable | -aenable | -arequire][[-pRepository] [-tFileType]: [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<i>-a</i>	Specify the defect attachment action on check in: <i>disable, enable, require</i>
<i>-b</i>	Enter the branch name. The default branch is set in the working directory.

<code>-d</code>	List deleted files in a repository.
<code>-i</code>	Inherit the parent repository TestTrack Pro configuration settings.
<code>-k</code>	Set keyword expansion: <i>on, off, default</i>
<code>-o</code>	Specifies if users can override the default TestTrack Pro database configuration. Enter <code>-o-</code> if you do not want to let users override the default.
<code>-p</code>	Enter the full path of the parent repository for a file or repository.
<code>-s</code>	Enter the TestTrack Pro database configuration name.
<code>-t</code>	Enter the file type: <i>auto_detect, text, binary, or mac_binary</i>

rebasefile

Update a file or repository in a child branch with changes made to a file or repository in the parent branch. Rebasing repositories to make sure you have the most recent copy of the files.

Shortcut command: **rf**

Syntax

```
sscm rebasefile Item1 [Item2 ...] [-bBranch] [-ccComment | -cpCommentPath | -c-] [-h] [-lLabel]
[-nSnapshotBranch] [-pRepository] [-r] [-s] [-tTimeStamp] [-v] [-yUsername:Password]
[-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the file or repository name. Can be / or empty, which means the repository specified by the <code>-p</code> option or the default repository.
<code>-b</code>	Enter the child branch name you want to update. The default branch is set in the working directory. Note: The <code>-b</code> , <code>-n</code> , and <code>-t</code> options are mutually exclusive. If one of these options is not specified, the current file version in the parent branch will be copied.
<code>-c</code>	Enter a comment or the comment text file path. Enter <code>-c</code> to prompt for comment input. Enter <code>-c-</code> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<code>-l</code>	Enter a label to determine the file version that will be copied from the parent branch to the child branch.
<code>-n</code>	Enter a snapshot branch to determine the file version that will be copied from the parent branch to the child branch.
<code>-p</code>	Enter the full repository path.
<code>-r</code>	Recursively rebase all files and sub-repositories.

<code>-s</code>	Skip the server auto-merge.
<code>-t</code>	Enter the timestamp to determine the file version that will be copied from the parent branch to the child branch. Use the following date/time format: <i>yyymmddhh:mm:ss</i>
<code>-v</code>	Preview the result of rebasing the changed files.

rename

Rename a file or repository.

Syntax

```
sscm rename CurrentName NewName [-pRepositoryPath [-bBranch]] [-ccComment | -cpCommentPath] [-h] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>CurrentName</i>	Enter the existing file or repository name.
<i>NewName</i>	Enter the new file or repository name.
<code>-b</code>	Enter the name of the branch.
<code>-p</code>	Enter the full path of the parent repository for a file or repository.

reporthistory

The history report provides historical information for a file or repository. You can generate a report based on specific actions, dates, users or a combination of all three.

Shortcut command: **rh**

Syntax

```
sscm reporthistory SubRepo [-aAction] [-bBranch] [-dDateTimeFrom-DateTimeTo] [-h] [-oReportFile] [-pRepository] [-r] [-uSearchUsername] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>SubRepo</i>	Enter the sub-repository name. Can be <code>/</code> , which means the sub-repository specified by the <code>-p</code> option or the default repository.
<code>-a</code>	Display the file history by only listing the specified actions. Actions include: AddToRepository, AddToBranch, AddFromBranch, CheckIn, Rebase, RebaseWithMerge, Promote, Label, AttachToDefect, Delete, Undelete, All and AllChanges (default). The default action (AllChanges) means any action that changes the version number of the file including add, check in, promote, rebase, and remove.
<code>-b</code>	Enter the name of the branch you want to run the report for. The default branch is set in the working directory.
<code>-d</code>	Display file information inclusively between the two specified days/times. Date/time format: <i>yyyymmddhh:mm:ss</i> . The default is an open-ended day/time range.

<code>-o</code>	Enter the name of the file to output information to. Report information is written to this file instead of being displayed on the screen.
<code>-p</code>	Enter the full path of the parent repository for a file or repository.
<code>-r</code>	Recursively run the report for all files and sub-repositories.
<code>-i</code>	Include all repositories in the report or only include repositories with found items.
<code>-u</code>	Enter a username to display changes made by a specific user. The default is all users.

restore

Restore a file or a repository which was deleted.

Syntax

```
sscm restore Item [-bBranch] [-ccComment | -cpCommentPath | -c-] [-h] [-pRepository] [-yUsername:Password]
[-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<code>-b</code>	Enter the branch name. The default branch is set in the working directory.
<code>-c</code>	Enter a comment or the comment text file path. Enter <code>-c</code> to prompt for comment input. Enter <code>-c-</code> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<code>-p</code>	Enter the full repository path for the file or the repository.

rm

Delete a file.

Syntax

```
sscm rm Item [-bBranch] [-ccComment | -cpCommentPath | -c-] [-f] [-d] [-h] [-pRepository]
[-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<code>-b</code>	Enter the branch name. The default branch is set in the working directory.

<code>-c</code>	Enter a comment or the comment text file path. Enter <code>-c</code> to prompt for comment input. Enter <code>-c-</code> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<code>-d</code>	Permanently destroy the file.
<code>-f</code>	Force removal of non-empty sub-repository.
<code>-p</code>	Enter the full repository path for the file or the repository.

rollback

Rollback files to revert to a previous file version without affecting file history and version numbers. Changes made to the file are discarded. The version number increments even though the contents are the same as a previous file version.

Shortcut command: **rbk**

Syntax

```
sscm rollback File [-bBranch] [-ccComment | -cpCommentPath | -c-] [-h] [-pRepository] -vVersion
[-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>File</i>	Enter the name of the file you want to rollback.
<code>-b</code>	Enter the name of the branch that contains the file.
<code>-c</code>	Enter a comment or the comment text file path. Enter <code>-c</code> to prompt for comment input. Enter <code>-c-</code> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<code>-p</code>	Enter the full path of the repository.
<code>-v</code>	Enter the version number to rollback to. This option is required.

share

Sharing files creates a link in one repository to a file in another repository within the same branch. This lets you maintain common code while eliminating the need to work with multiple copies of the same source file. Operations on the shared file are applied to the base (linked to) file. When a file is shared, a shared history and filename are maintained. Only the file location is different.

Syntax

```
sscm share SourceRepository [File1 File2 ...] [-bBranch] [-ccComment | -cpCommentPath | -c-] [-h]
[-pDestinationRepository] [-r] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>SourceRepository</i>	Full path of the source repository.
<i>File</i>	A file or list of files to be shared. Can be empty, which means share the source repository.
<i>-b</i>	Enter the branch name you want to share files in. The default branch is set in the working directory.
<i>-p</i>	Enter the full path of the destination repository.
<i>-r</i>	Recursively share all files and sub-repositories.

uncheckout

Cancel a checkout and disregard all changes made after checkout.

Shortcut command: **uco**

Syntax

```
sscm uncheckout Item1 [Item2...] [-bBranch] [-flatest | -foriginal | -fleave] [-h] [-pRepository] [-q] [-r]
[-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<i>-b</i>	Enter the branch name. The default branch is set in the working directory.
<i>-f</i>	Enter a file overwrite option: <i>latest</i> (default), <i>original</i> , <i>leave</i>
<i>-p</i>	Enter the full repository path.
<i>-q</i>	Quiet mode. Do not list repository and local full path of the Surround SCM server.
<i>-r</i>	Recursively uncheckout all files and sub-repositories.

voidcheckout

Void a user's check out operation. This breaks the locks on the selected files, giving other users access to the files. Any changes the user made to the files are not saved.

Shortcut command: **vco**

Syntax

```
sscm voidcheckout Username Item1 [Item2...] [-bBranch] [-h] [-pRepository] [-yUsername:Password]
[-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<i>-b</i>	Enter the branch name of the of the Surround SCM server or repositories.
<i>-p</i>	Enter the full path of the parent repository.
<i>-r</i>	Recursively void checkout for all files and sub-repositories.

Repository commands

A repository is a collection of files and/or subrepositories that helps you organize Surround SCM. Each repository is generally mapped to a directory on your hard drive. You can create as many repositories as necessary. Your company's needs and business processes dictate how you set up repositories.

diff

View differences between the file in your working directory and a version of the file or between two versions of the file.

Syntax

```
sscm diff File [-bBranch1[:Branch2]] [-h] [-pRepository] [-uDiffUtility] [-vVersion1[:Version2]]
[-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>File</i>	Enter the file name.
<i>-b</i>	Enter the branch name. If a branch is not specified, the default branch is used. You can compare files between the working directory and a branch, file versions within the same branch, or files between two branches.
<i>-p</i>	Enter the full repository path.
<i>-u</i>	Enter the 3rd party diff utility you want to use. You must include the full path of the diff utility package, %1 for the position of the first file and %2 for the position of the second file, and/or other options of the utility. The default utility is used if an option is not specified.
<i>-v</i>	Display the difference between the two versions.

history

Show the history of a repository to view date, user, action performed and file version number.

Syntax

```
sscm history Item [-aAction] [-bBranch] [-dDateFrom:DateTo] [-h] [-pRepository] [-uUsername]
[-vVersionFrom:VersionTo] [-yUsername:password] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<i>-a</i>	Displays the history by only listing the specified action. Possible actions include: AddToRepository, AddToBranch, AddFromBranch, CheckIn, Rebase, RebaseWithMerge, Promote, Label, AttachToDefect, Delete, Undelete
<i>-b</i>	Enter the branch name. The default branch is set in the working directory.
<i>-d</i>	Display the file history between two specified dates. Date format: <i>yyyymmdd</i>
<i>-p</i>	Enter the full repository path.
<i>-u</i>	Display changes made by a specific user.
<i>-v</i>	Display changes between the two versions.

label

Labels provide a way to mark a specific version of files in a repository. When a label is created, a new entry is added to the repository history. The file, and the version number, do not change. For example, you can label a repository that contains all the files for a testing release. Every time you release the build to testing, create a corresponding label. You can view the history to see what changed after a label was created. You can also easily recompile a specific version of the testing build.

Syntax

```
sscm label Item1 [Item2 ...] [-bBranch] [-ccComment | -cpCommentPath | -c-] [-h] [-lLabel] [-o] [-pRepository]
[-r] [-vVersion] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<i>-b</i>	Enter the branch name. The default branch is set in the working directory.
<i>-c</i>	Enter a comment or the comment text file path. Enter <i>-c</i> to prompt for comment input. Enter <i>-c-</i> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<i>-l</i>	Enter the new label. To prompt for input, do not enter an <i>-l</i> option.

<code>-o</code>	Overwrite the existing label.
<code>-p</code>	Enter the full path of the parent repository.
<code>-r</code>	Recursively label all files.
<code>-v</code>	Enter the version number of a file.

merge

View differences between server files and working directory files and merge changes into one file.

Syntax

`sscm merge File [-bBranch] [-h]`

`[-pRepository] [-uMergeUtility] [-vVersion] [-yUsername:Password] [-zSCMServerAddr:PortNum]`

<i>File</i>	Enter the file name.
<code>-b</code>	Enter the branch name. If a branch is not specified, the default branch is used.
<code>-p</code>	Enter the full repository path.
<code>-u</code>	Enter the 3-way merge utility you want to use. You must include the full path of the merge utility package, %1 for the position of the common version file, %2 for the position of the new version file, %3 for the position of the local changed file, and/or other options of the utility. The default utility is used if an option is not specified.
<code>-v</code>	Merge from the specified version.

mkrepository

Create a repository in a branch.

Shortcut command: **mkrepo**, **mr**

Syntax

`sscm mkrepository SubRepository [-bBranch] [-ccComment | -cpCommentPath | -c-] [-h] [-pRepository] [-yUsername:Password] [-zSCMServerAddr:PortNum]`

<code>-b</code>	Enter the branch name. The default branch is set in the working directory.
<code>-c</code>	Enter a comment or the comment text file path. Enter <code>-c</code> to prompt for comment input. Enter <code>-c-</code> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<code>-p</code>	Enter the full repository path.

promotefile

Update a repository in an ancestor branch with file or repository changes in a child branch. You can promote all files in a repository without affecting other items in the branch.

Shortcut command: **pf**

Syntax

```
sscm promotefile Item1 [Item2 ...] [-bBranch] [-ccComment | -cpCommentPath | -c-] [-h] [-pRepository] [-r] [-s]
[-uUpperBranch] [-v] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the file or repository name. Can be / or empty, which means the repository specified by the -p option or the default repository.
<i>-b</i>	Enter the name of the child branch you want to promote changes from. The default branch is set in the working directory.
<i>-c</i>	Enter a comment or the comment text file path. Enter -c to prompt for comment input. Enter -c- if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<i>-p</i>	Enter the full repository path.
<i>-r</i>	Recursively promote all files and sub-repositories.
<i>-s</i>	Skip the server auto-merge.
<i>-u</i>	Enter the ancestor branch to be updated. This branch can be the parent branch, a grandparent branch, etc. If this option is not specified, the parent branch is used.
<i>-v</i>	Preview the results of promoting the changed files.

property

Set, then display, repository properties.

Syntax

```
sscm property Item [-bBranch] [-d] [-kon | -koff | -kdefault] [-h] [-i] | [-o | -o- -sTTDbConfiguration -
adisable | -aenable | -arequire] [-pRepository] [-tFileType] [-yUsername:Password]
[-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<i>-a</i>	Enter the defect attachment action on check in: <i>disable</i> , <i>enable</i> , <i>require</i> .
<i>-b</i>	Enter the branch name. The default branch is set in the working directory.

<code>-d</code>	List deleted files in a repository.
<code>-i</code>	Inherit the parent repository TestTrack Pro configuration settings.
<code>-k</code>	Set keyword expansion: <i>on</i> , <i>off</i> , or <i>default</i> .
<code>-o</code>	Specifies if users can override the default TestTrack Pro database configuration. If you do not want to let users override the default, enter <code>-o-</code> .
<code>-p</code>	Enter the full path of the parent repository for a file or repository.
<code>-s</code>	Enter the TestTrack Pro server configuration name.
<code>-t</code>	Enter the file type: <i>auto_detect</i> , <i>text</i> , <i>binary</i> and <i>mac_binary</i> .

rebasefile

Update a repository in a child branch with changes made to a file or repository in the parent branch. Rebasing repositories to make sure you have the most recent copy of the files, including changes made by other users.

Shortcut command: **rf**

Syntax

```
sscm rebasefile Item1 [Item2 ...] [-bBranch] [-ccComment | -cpCommentPath | -c-] [-h] [-lLabel]
[-nSnapshotBranch] [-pRepository] [-r] [-s] [-tTimeStamp] [-v] [-yUsername:Password]
[-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the file or repository name. Can be <code>/</code> or empty, which means the repository specified by the <code>-p</code> option or the default repository.
<code>-b</code>	Enter the name of the child branch you want to update. The default branch is set in the working directory. Note: The <code>-b</code> , <code>-n</code> , and <code>-t</code> options are mutually exclusive. If one of these options is not specified, the current file version in the parent branch will be copied.
<code>-c</code>	Enter a comment or the comment text file path. Enter <code>-c</code> to prompt for comment input. Enter <code>-c-</code> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<code>-l</code>	Enter a label to determine the file version that will be copied from the parent branch to the child branch.
<code>-n</code>	Enter a snapshot branch to determine the file version that will be copied from the parent branch to the child branch.
<code>-p</code>	Enter the full repository path.
<code>-r</code>	Recursively rebase all files and sub-repositories.

<code>-s</code>	Skip the server auto-merge.
<code>-t</code>	Enter the timestamp to determine the file version that will be copied from the parent branch to the child branch. Use the following date/time format: <i>yyymmddhh:mm:ss</i>
<code>-v</code>	Preview the result of rebasing the changed files.

rename

Rename a repository or file.

Syntax

`sscm rename CurrentName NewName [-pRepositoryPath [-bBranch]] [-ccComment | -cpCommentPath] [-h] [-yUsername:Password] [-zSCMServerAddr:PortNum]`

<i>CurrentName</i>	Enter the existing file or repository name.
<i>NewName</i>	Enter the new file or repository name.
<code>-b</code>	Enter the name of the branch.
<code>-p</code>	Enter the full path of the parent repository for a file or repository.

restore

Restore a deleted file or a repository.

Syntax

`sscm restore Item [-bBranch] [-ccComment | -cpCommentPath | -c-] [-h] [-pRepository] [-yUsername:Password] [-zSCMServerAddr:PortNum]`

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<code>-b</code>	Enter the branch name. The default branch is set in the working directory.
<code>-c</code>	Enter a comment or the comment text file path. Enter <code>-c</code> to prompt for comment input. Enter <code>-c-</code> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<code>-p</code>	Enter the full repository path.

rm

Delete a repository.

Syntax

```
sscm rm Item [-bBranch] [-ccComment | -cpCommentPath | -c-] [-f] [-d] [-h] [-pRepository]
[-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>Item</i>	Enter the repository name. Can be / or empty, which means the repository specified by the repository option or the default repository.
<i>-b</i>	Enter the branch name. The default branch is set in the working directory.
<i>-c</i>	Enter a comment or the comment text file path. Enter <i>-c</i> to prompt for comment input. Enter <i>-c-</i> if a comment is not needed. CommentPath can be the full path of a local text file, partial path if the file is in a subdirectory of the current directory, or the file name if the file is in the current directory.
<i>-d</i>	Permanently destroy the repository.
<i>-f</i>	Force removal of non-empty sub-repository.
<i>-p</i>	Enter the full repository path.

rmworkdir

Remove a working directory.

Shortcut command: **rmwd**

Syntax

```
sscm rmworkdir [-pRepository [-bBranch | -iBranchID]] [-cComputer] [-h] [-yUsername:Password]
[-zSCMServerAddr:PortNum]
```

<i>-p</i>	The repository the working directory is defined for.
<i>-b</i>	The branch the working directory is defined for. This argument cannot be used without also specifying the repository (-p) argument. This argument cannot be used if the -i argument is specified.
<i>-i</i>	The branch id the working directory is defined for. If the branch name cannot be determined, use the branch id from the showworkdirs command. You must also specify the repository (-p) argument. This argument cannot be used if the -b argument is specified.
<i>-c</i>	The computer the working directory is specified for. This is the computer listed from the showworkingdirs command.

securerepository

Set, then display, security for the selected repository. You can apply security at the repository level, overriding the server level security. Security can be modified for any group for the selected repository. If you only want to view the repository security, do not specify `-a`, `-f`, or `-r` options.

Shortcut command: **sr**

Syntax

```
sscm securerepository [-a | -a-] [-bBranch | -o] [-f±Command1[±Command2 ...] [-gGroupName] [-h]
[-pRepository] [-r] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

-a	Apply security group to the repository. If the group already exists and inherits security, copy the inherited security settings to the repository and mark the group security as not inherited. If OtherGroups does not exist, the group cannot be removed and will inherit security.
-b	Enter the branch name. If a branch is not specified, the default branch is used.
-f	Enter the commands to be added or deleted. The following file commands can be added/deleted for repository security: RepoListVisible, Add, Remove, Get, CheckIn, ChangeType, ChangeProperties, History, ShareFiles, BreakShares, and All.
-g	Enter the security group name.
-o	Set or list the overall security settings for the repository.
-p	Enter the full repository path.
-r	Recursively force all child repositories to inherit these security settings.

Security group commands

addgroup

Add a new security group.

Shortcut command: **ag**

Syntax

```
sscm addgroup GroupName [-dDescription] [-h] [-nNote]
[-sCategory1±Command1[:Category2±Command2...]][-u±Username1[:±Username2...]]
[-yUsername:Password] [-zSCMServerAddr:PortNum]
```

-d	Enter the security group description.
-n	Enter notes about the security group.

-s	Enter command security information. Add to, or delete from, the group the command(s) in the category/categories.
-u	Enter which user(s) will be added to, or deleted from, the security group.

editgroup

Edit an existing security group.

Shortcut command: **eg**

Syntax

```
sscm editgroup GroupName [-dDescription] [-h] [-nNote]
[-sCategory1±Command1[:Category2±Command2...]] [-u±Username1[:±Username2...]]
[-yUsername:Password] [-zSCMServerAddr:PortNum]
```

-d	Enter the security group description.
-n	Enter the security group notes.
-s	Enter the command security information. Add to, or delete from, the group the command(s) in the category/categories.
-u	Enter which user(s) will be added into or deleted from the security group.

lsgroup

Display information of security group(s) in the SCM server.

Shortcut command: **lg**

Syntax

```
sscm lsgroup [GroupName] [-d] [-fBranch1:Repository1 [-fBranch2:Repository2]] [-h] [-n]
[-s[Category1[:Category2...]]] [-u] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

-d	Display the security group description.
-f	Enter the branch name and repository name to list the security information for the group.
-n	Display the security group notes.
-s	Display all enabled commands in the category/categories.
-u	Display all users in the security group.

rmgroup

Delete a security group.

Shortcut command: **rg**

Syntax

sscm rmgroup *GroupName* [-h] [-y*Username:Password*] [-zSCMServerAddr:PortNum]

Server commands

editmailqueue

Edit the status of pending messages in the server's mail queue.

Abbreviated command: **emq**

Syntax

sscm editmailqueue *MessageID* | -a -r | -d [-h] [-y*Username:Password*] [-zSCMServerAddr:PortNum]

-a :	Apply the action to all pending messages.
-r :	Reattempt the selected message(s).
-d :	Delete the selected message(s).

lsmailqueue

Lists pending messages in the server's mail queue.

Shortcut command: **lsmq**

Syntax

sscm lsmailqueue [-h] [-y*Username:Password*] [-zSCMServerAddr:PortNum]

lsserverlog

Display the server log.

Shortcut command: **lsl**

Syntax

sscm lsserverlog [-a] [-f*Field*] [-h] [-l*Level*] [-p*PageNumber*] [-s*Size*] [-t*Type*] [-y*Username:Password*] [-zSCMServerAddr:PortNum]

-a	Sort by ascending order. The default is descending.
----	---

-f	Enter the field (column) number to sort. The field numbers are defined as: 1. date/time (default) 2. level 3. error number 4. message string 5. mainline branch name 6. username
-l	Enter the log level number to server database. The log levels are defined as: 1. severe error 2. error 3. warning 4. unusual activity 5. information (default) 6. debug information (debug build only)
-p	Enter the page number. The default is 1.
-s	Enter the page size, number of records on one page. The default is 20.
-t	Enter the filter type to server log. The filter types are defined as: 1. specified level and more severe (default) 2. specified level and less severe 3. specified level only

rmserverlog

Delete the server log.

Shortcut command: **rsi**

Syntax

sscm rmserverlog -a | -t*DateTime* [-h] [-y*Username:Password*] [-z*SCMServerAddr:PortNum*]

-a	Delete all log records. One and only one of -a and -t must be specified.
-t	Enter a timestamp. Log records older than the timestamp are deleted. Date/Time format: <i>yyyymmddhh:mm:ss</i>

serveremailoption

Set then display server email options. To view server settings, enter this command without specifying any options.

Shortcut command: **seo**

Syntax

sscm serveremailoption [-h] [-m | -m-] [-x | -x-] [-rProfileName] [-wPassword] [-s | -s-] [-v | -v-] [-o | -o-] [-tSMTPHost] [-uUsername] [-pPassword] [-a | -a-] [-nName] [-eEmailAddress] [-yUsername:Password] [-zSCMServerAddr:PortNum]

-m	Enable sending of mail via MS Mail/Exchange (MAPI) with -m or disable it with -m-.
-x	Pause sending of mail via MAPI with -x or unpause with -x-.
-r	Enter the MAPI profile name used to send mail.
-w	Enter the password for the MAPI profile.
-s	Enable sending of mail via SMTP with -s or disable it with -s-.
-v	Pauses sending of mail via SMTP with -v or unpause with -v-.
-o	Enter -o to only send one message per SMTP connection. Enter -o- to send as many emails as possible on the open SMTP connection.
-t	Enter the SMTP host to send mail through.
-u	Enter the username for SMTP if authentication is required.
-p	Enter the password for the SMTP username.
-a	Always use the notification email account for the return address. -a- only uses the notification account email if the logged in user did not provide an email address.
-n	Enter the notification account name.
-e	Enter the notification account email address.

serveroption

Set then display the server options. To view server settings, enter this command without any options.

Shortcut command: **so**

Syntax

```
sscm serveroption [-aRAM] [-c | -c-] [-e | -e-] [-h] [-i | -i-] [-k | -k-][LicenseServerAddr:PortNum] [-m | -m-]
[-nLevel] [-pLicenseServerPassword] [-oLevel][qAddress][-ji | -jm][-fAddress][-r<o-4096>]
[-sLevel][tCacheRefreshRate] [-uSCMAdminPassword][-w | -w-][-yUsername:Password]
[-zSCMServerAddr:PortNum]
```

-a	Enter the amount of memory Guiffy can use on the server. The recommended setting is 128 MB. This setting should not exceed 50% of the server computer's physical memory. Depending on the amount of available RAM, you may want to set this lower. If you set the amount of memory too high, other applications can be negatively affected.
-c	Set compression of files in Surround SCM on with -c or off with -c-.
-e	Set encryption of communications on with -e or off with -e-.
-f	Change the email return address for error notifications.
-i	Specifies if information should be written to the startup log file.
-j	Specifies the type of email. Enter -ji to specify an Internet address. Enter -jm to specify a MAPI address.
-k	Set keyword expansion on with -k or off with -k-.
-l	Set the license server address and port number.
-m	Set multiuser checkouts on with -m or off with -m-.
-n	Enter the NT log level number. The log levels are defined as: 0. no message logged 1. severe error 2. error 3. warning 4. unusual activity 5. information (default) 6. debug information (debug build only)

<code>-o</code>	Enter the email notification log level number. The log levels are defined as: 0. no message logged 1. severe error 2. error 3. warning 4. unusual activity 5. information (default) 6. debug information (debug build only)
<code>-p</code>	Enter the license server communication password.
<code>-q</code>	Enter the email address to send error notifications to.
<code>-r</code>	To require a minimum comment length, enter the minimum number of characters. This option applies to all commands with optional comment fields, such as check in or rebase.
<code>-s</code>	Enter the log level number to NT log. The log levels are defined as: 0. no message logged 1. severe error 2. error 3. warning 4. unusual activity 5. information (default) 6. debug information (debug build only)
<code>-t</code>	Set the cached user list refresh rate (in minutes).
<code>-u</code>	Set the local Surround SCM admin password.
<code>-w</code>	Specify if a warning should be generated if the server is not started as a NT service.

setclient

Set then display client options.

Shortcut command: **sc**

Syntax

```
sscm setclient [-dDiffUtility] [-h] [-mMergeUtility] [-w | -w-] [-c | -c-] [-i | -i-] [-jPath] [-gPath] [-n | -n-]
[-r | -r-] [-aRAM] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<code>-d</code>	Enter the diff utility to be used by command diff. Must include the full path of the diff utility, %1 for the position of first file and %2 for the position of the second file, and/or other options for the utility. The default utility is used if an option is not specified.
<code>-m</code>	Enter the 3rd party 3-way merge utility provided by user. It must include the full path of the merge utility package, %1 for the position of the common version file, %2 for the position of the newer version file, %3 for the position of the local changed file, and/or other options of the utility. The default utility is used if an option is not specified.

-w	Specifies files are in Windows text format. -w for Windows and -w- for Unix.
-c	Set data compression during file transfer on with -c or off with -c-.
-i	Use the integrated Guiffy diff/merge utility with -i or use selected diff/merge utilities with -i-.
-j	Set the Java runtime path for use with the Guiffy diff/merge utility.
-g	Set the path for use with the Guiffy diff/merge utility.
-n	Set option to update version even if no change is made on with -n or off with -n-.
-r	Set local file removal on check in on with -r or off with -r-.
-a	Enter the amount of memory Guiffy can use. The recommended setting is 128 MB. This setting should not exceed 50% of the computer's physical memory. Depending on the amount of available RAM, you may want to set this lower.

version

List the versions of the server and command line application.

Syntax

sscm version [-h] [-yUsername:Password] [-zSCMServerAddr:PortNum]

TestTrack Pro integration commands

Surround SCM’s two-way, seamless integration with TestTrack Pro provides complete change management by tracking what changed and why. This integration lets you link changes with defects, feature requests, and change requests. Files can be viewed from Surround SCM or TestTrack Pro.

fetchttldb

List the databases stored on the TestTrack Pro server.

Shortcut command: **ftt**

Syntax

sscm fetchttldb TTProServerAddress:PortNum [-h] [-ySCMUsername:SCMPassword]
[-zSCMServerAddr:PortNum]

TTProServerAddress:PortNum	Enter the TestTrack Pro server address and port number.
----------------------------	---

lsdefect

List the defects assigned to a TestTrack Pro user.

Shortcut command: **ld**

Syntax

sscm lsdefect *TTDbConfigurationName* [-h] [-lTTUsername:TTPassword][
-ySCMUsername:SCMPassword]
[-zSCMServerAddr:PortNum]

<i>TTDbConfigurationName</i>	Enter the TestTrack database configuration name.
-l	Enter the TestTrack username and password.

lsttdb

List the TestTrack Pro database configurations.

Shortcut command: **ltt**

Syntax

sscm lsttdb [-h] [-ySCMUsername:SCMPassword] [-zSCMServerAddr:PortNum]

rmttddb

Delete a TestTrack Pro database configuration.

Shortcut command: **rtt**

Syntax

sscm rmttddb *TTProDbConfigName* [-h] [-yUsername:Password] [-zSCMServerAddr:PortNum]

<i>TTProDbConfigName</i>	Enter the name of the database configuration you want to delete.
--------------------------	--

setttddb

Add a TestTrack Pro database configuration or edit an existing configuration.

Shortcut command: **stt**

Syntax

sscm setttddb *TTProDbConfigName* *TTProServerAddress:PortNum* *TTProDbId* [-h] [-i] [-yUsername:Password]
[-zSCMServerAddr:PortNum]

<i>TTProDbConfigName</i>	Enter a new TestTrack Pro database configuration name or enter an existing name for editing. The name is unique on Surround SCM server.
--------------------------	---

<i>TTProDbId</i>	Enter the TestTrack Pro database id number. This id number can be found using the <i>fetchtttd</i> command.
<i>-i</i>	Initialize the setting for a new configuration.

Trigger commands

Surround SCM includes pre- and post-event triggers that are stored on the Surround SCM server. Triggers can only be fired from events on files. Each trigger is run once per file that it is associated with.

Pre-event triggers run after a client requests that a command be performed, but before the event is complete. These trigger types let a server script perform additional checks on a file before the event completes. Pre-event triggers are for validation and custom text entry, such as keyword expansion. For example, you can create scripts for custom keyword expansion or to verify that text, such as a copyright notice, is included in certain files before check in. Post-event triggers run after a command is successfully completed on the server. These triggers can either run a script or send email. Post-event triggers are for logging and synchronization. For example you can create scripts for custom logging of SCM file activity or to synchronize two separate Surround SCM servers.

Email triggers use a standard email template, which can be customized, to inform selected users when an event occurs to a file or a set of files. Each email template is stored with the trigger on the server.

Triggers can attach to the following file events: (all), (file version updated), (new file), Add, Add from branch, Attach to defect, Check in, Promote from, Rebase, Remove, Rename file, Restore, Share broken, Share created.

Note: Post-event triggers should always be used unless the trigger is being used for validation purposes. Triggers, especially pre-event, can adversely affect performance. Pre-event triggers are run once per file and the Surround SCM server waits for the script to complete before moving to the next file. Even a one second pre-event script will significantly slow down the server because it will take one additional second per file that the trigger fires on. Post-event triggers may impact performance because each file affected by the action must be checked for existing triggers.

addemailnotification

Add an email notification.

Shortcut command: **aen**

Syntax

```
sscm addemailnotification [-h] [-dDescription] [-bBranch] -pRepository [-r | -r-] -fFileMatch [-e | -e-]
[-c | -c-] -aEvent [-x | -x-] [-o | -o-] [-n | -n-] [-uSubject] [-tTemplateFile] [-lFileListFile] [-i | -i-] [-j | -j-]
[-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>-d</i>	Enter the notification description.
-----------	-------------------------------------

<code>-b</code>	Enter the branch the trigger should act on. If a branch is not specified, the trigger acts on all branches.
<code>-p</code>	Enter the repository for the trigger to act on.
<code>-r</code>	Enter <code>-r</code> for a recursive repository search or <code>-r-</code> for non-recursive.
<code>-f</code>	Enter the files to match.
<code>-e</code>	Enter <code>-e</code> to specify the file match is a regular expression. Enter <code>-e-</code> to specify the file match is a wildcard search.
<code>-c</code>	Enter <code>-c</code> to specify the file match is case sensitive search. Enter <code>-c-</code> to specify the match is case independent.
<code>-a</code>	Specifies one of the following actions: all, fileversionupdated, newfile, add, addtobranh, addfrombranch, checkin, rebase, rebasewithmerge, promoteto, promotefrom, attachtodeflect, rm, restore, sharecreated, sharebroken, rename, destroy.
<code>-x</code>	Enter <code>-x</code> to enable the notification or <code>-x-</code> to disable it.
<code>-o</code>	Enter <code>-o</code> to disable the notification after it runs once.
<code>-n</code>	Enter <code>-n</code> to consolidate notifications that act on more than one file into one email. Enter <code>-n-</code> to specify one email be sent per file.
<code>-u</code>	Enter the email notification subject.
<code>-t</code>	Enter the email template filename.
<code>-l</code>	Enter the %SSCM_FILELIST% template filename.
<code>-i</code>	Enter <code>-i</code> to send each email to a single recipient or <code>-i-</code> to send each email to multiple recipients.
<code>-j</code>	Enter <code>-j</code> to send email in HTML format or <code>-j-</code> to send email in plain text format.

addtrigger

Add a trigger.

Shortcut command: **at**

Syntax

```
sscm addtrigger [-h] [-dDescription] [-bBranch] -pRepository [-r | -r-] -fFileMatch [-e | -e-] [-c | -c-] -wbefore | -
  wafter -aEvent [-x | -x-] [-o | -o-] -sScript | -mUsers [-kAddresses] [-n | -n-] [-uSubject] [-tTemplateFile] [-
  lFileListFile] [-i | -i-] [-j | -j-] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<code>-d</code>	Enter the trigger description.
-----------------	--------------------------------

<code>-b</code>	Enter the branch the trigger should act on. If a branch is not specified, the trigger acts on all branches.
<code>-p</code>	Enter the repository for the trigger to act on.
<code>-r</code>	Enter <code>-r</code> for a recursive repository search or <code>-r-</code> for non-recursive.
<code>-f</code>	Enter the files to match.
<code>-e</code>	Enter <code>-e</code> to specify the file match is a regular expression. Enter <code>-e-</code> to specify the file match is a wildcard search.
<code>-c</code>	Enter <code>-c</code> to specify the file match is case sensitive search. Enter <code>-c-</code> to specify the match is case independent.
<code>-w</code>	Enter <code>-wbefore</code> to specify a pre-event trigger. Enter <code>-wafter</code> to specify a post-event trigger.
<code>-a</code>	Specifies one of the following actions: all, fileversionupdated, newfile, add, addtobranch, addfrombranch, checkin, rebase, rebasewithmerge, promoteto, promotefrom, attachtodeflect, rm, restore, sharecreated, sharebroken, rename, destroy.
<code>-x</code>	Enter <code>-x</code> to enable the trigger or <code>-x-</code> to disable it.
<code>-o</code>	Enter <code>-o</code> to disable the trigger after it runs once.
<code>-s</code>	Enter the script the trigger should run.
<code>-m</code>	Enter the SCM usernames the trigger should send mail to.
<code>-k</code>	Enter the email addresses the trigger should send mail to.
<code>-n</code>	Enter <code>-n</code> to consolidate triggers that act on more than one file into one email. Enter <code>-n-</code> to specify one email be sent per file.
<code>-u</code>	Enter the email subject.
<code>-t</code>	Enter the email template filename.
<code>-l</code>	Enter the %SSCM_FILELIST% template filename.
<code>-i</code>	Enter <code>-i</code> to send each email to a single recipient or <code>-i-</code> to send each email to multiple recipients.
<code>-j</code>	Enter <code>-j</code> to send email in HTML format or <code>-j-</code> to send email in plain text format.

editemailnotification

Edit an email notification.

Shortcut command: **een**

Syntax

```
sscm editemailnotification [-h] TriggerID [-dDescription] [-bBranch] [-pRepository] [-r | -r-] [-fFileMatch]
[-e | -e-] [-c | -c-] [-aEvent] [-x | -x-] [-o | -o-] [-n | -n-] [-uSubject] [-tTemplateFile] [-lFileListFile] [-i | -i-]
[-j | -j-] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

-TriggerID	Numeric ID of the notification you want to edit. Use the lsemailnotification command to find the ID.
-d	Enter the notification description.
-b	Enter the branch the trigger should act on. If a branch is not specified, the trigger acts on all branches.
-p	Enter the repository for the trigger to act on.
-r	Enter -r for a recursive repository search or -r- for non-recursive.
-f	Enter the files to match.
-e	Enter -e to specify the file match is a regular expression. Enter -e- to specify the file match is a wildcard search.
-c	Enter -c to specify the file match is case sensitive search. Enter -c- to specify the match is case independent.
-a	Specifies one of the following actions: all, fileversionupdated, newfile, add, addtobranch, addfrombranch, checkin, rebase, rebasewithmerge, promoteto, promotefrom, attachtodefekt, rm, restore, sharecreated, sharebroken, rename, destroy.
-x	Enter -x to enable the notification or -x- to disable it.
-o	Enter -o to disable the notification after it runs once.
-n	Enter -n to consolidate notifications that act on more than one file into one email. Enter -n- to specify one email be sent per file.
-u	Enter the email notification subject.
-t	Enter the email template filename.
-l	Enter the %SSCM_FILELIST% template filename.

<code>-i</code>	Enter <code>-i</code> to send each email to a single recipient or <code>-i-</code> to send each email to multiple recipients.
<code>-j</code>	Enter <code>-j</code> to send email in HTML format or <code>-j-</code> to send email in plain text format.

edittrigger

Edit an existing trigger.

Shortcut command: **et**

Syntax

```
sscm edittrigger [-h] TriggerID [-dDescription] [-bBranch] [-pRepository] [-r | -r-] [-fFileMatch] [-e | -e-] [-c | -c-] [-wbefore | -wafter] [-aEvent] [-x | -x-] [-o | -o-] [-sScript] [-mAddresses] [-n | -n-] [-uSubject] [-tTemplateFile] [-lFileListFile] [-i | -i-] [-j | -j-] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<code>-TriggerID</code>	Numeric ID of the trigger you want to edit. Use the <code>lstrigger</code> command to find the ID.
<code>-d</code>	Enter the trigger description.
<code>-b</code>	Enter the branch the trigger should act on. If a branch is not specified, the trigger acts on all branches.
<code>-p</code>	Enter the repository for the trigger to act on.
<code>-r</code>	Enter <code>-r</code> for a recursive repository search or <code>-r-</code> for non-recursive.
<code>-f</code>	Enter the files to match.
<code>-e</code>	Enter <code>-e</code> to specify the file match is a regular expression. Enter <code>-e-</code> to specify the file match is a wildcard search.
<code>-c</code>	Enter <code>-c</code> to specify the file match is case sensitive search. Enter <code>-c-</code> to specify the match is case independent.
<code>-w</code>	Enter <code>-wbefore</code> to specify a pre-event trigger. Enter <code>-wafter</code> to specify a post-event trigger.
<code>-a</code>	Specifies one of the following actions: all, fileversionupdated, newfile, add, addtobranch, addfrombranch, checkin, rebase, rebasewithmerge, promoteto, promotefrom, attachtodefekt, rm, restore, sharecreated, sharebroken, rename, destroy.
<code>-x</code>	Enter <code>-x</code> to enable the trigger or <code>-x-</code> to disable it.
<code>-o</code>	Enter <code>-o</code> to disable the trigger after it runs once
<code>-s</code>	Enter the script the trigger should run.
<code>-m</code>	Enter the SCM usernames the trigger should send mail to.

<code>-k</code>	Enter the email addresses the trigger should send mail to.
<code>-n</code>	Enter <code>-n</code> to consolidate triggers that act on more than one file into one email. Enter <code>-n-</code> to specify one email be sent per file.
<code>-u</code>	Enter the email subject.
<code>-t</code>	Enter the email template filename.
<code>-l</code>	Enter the %SSCM_FILELIST% template filename.
<code>-i</code>	Enter <code>-i</code> to send each email to a single recipient or <code>-i-</code> to send each email to multiple recipients.
<code>-j</code>	Enter <code>-j</code> to send email in HTML format or <code>-j-</code> to send email in plain text format.

lsemailnotification

List all email notifications for the logged in user. Specify the TriggerID to view details about a specific notification.

Shortcut command: **len**

Syntax

sscm lsemailnotification [-h] [TriggerID] [-yUsername:Password] [-zSCMServerAddr:PortNum]

<code>-TriggerID</code>	Numeric ID of the email notification you want to view details about.
-------------------------	--

lstrigger

Lists all triggers. Specify the TriggerID to view details about a specific trigger.

Shortcut command: **lt**

Syntax

sscm lstrigger [-h] [TriggerID] [-yUsername:Password] [-zSCMServerAddr:PortNum]

<code>-TriggerID</code>	Numeric ID of the trigger you want to view details about.
-------------------------	---

rmemailnotification

Remove an email notification.

Shortcut command: **ren**

Syntax

sscm rmemailnotification [-h] *TriggerID* [-y*Username:Password*] [-zSCMServerAddr:PortNum]

<i>-TriggerID</i>	Numeric ID of the trigger you want to remove. Use the lsemailnotification command to find the ID.
-------------------	---

rmtrigger

Remove a trigger.

Shortcut command: **rt**

Syntax

sscm rmtrigger [-h] *TriggerID* [-y*Username:Password*] [-zSCMServerAddr:PortNum]

<i>-TriggerID</i>	Numeric ID of the trigger you want to remove. Use the lstrigger command to find the ID.
-------------------	---

User commands

adduser

Add a new user.

Shortcut command: **au**

Syntax

sscm adduser Username [-c*CompanyName[:Address]*] [-e*EmailType:Address*] [-f*FirstName*] [-h] [-i*floating | inamed | inone*] [-l*LastName*] [-n*Note*] [-p*PhoneType1:Number1[:PhoneType2:Number2]*] [-tTT*Username:TTPassword*] [-w*UserPasswd*] [-y*Username1:Password1*] [-zSCMServerAddr:PortNum]

<i>Username</i>	Enter the user name.
<i>-c</i>	Enter the company name and address.
<i>-e</i>	Enter the email type and address. Email type can be Internet, MAPI, or other.
<i>-f</i>	Enter the first name of the user.
<i>-i</i>	Specify the license type for the user.
<i>-l</i>	Enter the last name of the user.

<i>-n</i>	Enter any notes.
<i>-p</i>	Enter phone types and numbers. Phone type can be Work, Fax, Home, Pager and Mobile.
<i>-t</i>	Enter the TestTrack Pro username and password.
<i>-w</i>	Enter the user's password. No password following will prompt for input.

edituser

Edit an existing user.

Shortcut command: **eu**

Syntax

sscm edituser *Username* [-c*CompanyName[:Address]*] [-e*EmailType:Address*] [-f*FirstName*] [-h] [-i*floating* | -i*named* | -i*none*] [-l*LastName*] [-n*Note*] [-p*PhoneType1:Number1[:PhoneType2:Number2]*] [-t*TTUsername:TTPassword*] [-w*UserPasswd*] [-y*Username1:Password1*] [-z*SCMServerAddr:PortNum*]

<i>Username</i>	Enter the user name.
<i>-c</i>	Enter the company name and address.
<i>-e</i>	Enter the email type and address. Email type can be Internet, MAPI, or other.
<i>-f</i>	Enter the first name of the user.
<i>-i</i>	Specify the license type for the user.
<i>-l</i>	Enter the last name of the user.
<i>-n</i>	Enter any notes.
<i>-p</i>	Enter phone types and numbers. Phone type can be Work, Fax, Home, Pager and Mobile.
<i>-t</i>	Enter the TestTrack Pro username and password.
<i>-w</i>	Enter the user's password. No password following will prompt for input.

lsuser

Display information of user(s) in the SCM server.

Shortcut command: **lu**

Syntax

sscm lsuser [*User*][-c][-e][-f] [-g][-h][-i] [-l] [-n] [-o] [-p] [-s] [-t] [-w] [-y*Username:Password*][-zSCMServerAddr:PortNum]

<i>User</i>	Enter a user name to only view that user's information.
-c	Display the user's company information.
-e	Display the user's email address.
-f	Display the full name of the user.
-g	Display the global License Server users that are not used by the Surround SCM server.
-i	Display the user's license type.
-l	Display if the user is licensed.
-n	Display the user notes.
-o	Display the overall information about the user.
-p	Display the user's phone number.
-s	Display the security groups the user is in.
-t	Display the TestTrack Pro username.
-w	Display the working directories of the user.

passwd

Change your password.

Shortcut command: **pw**

Syntax

sscm passwd [-h] [-u*Username*] [-zSCMServerAddr:PortNum]

-u	Enter the username for the password being changed. This option is only needed if the user is not the default user on the computer.
----	--

retrieveuser

Retrieves global users, who reside on the license server, to provide them access to Surround SCM. If the user is assigned a named or floating license, they can also login and work with Surround SCM.

Shortcut command: **rtu**

Syntax

```
sscm retrieveuser Username1 [Username2 ...] [-h] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

rmuser

Delete a user.

Shortcut command: **ru**

Syntax

```
sscm rmuser Username [-h] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>Username</i>		Enter the user name you want to delete.
-----------------	--	---

Working directory commands

showworkdirs

Shows working directories in use for the specified computer.

Shortcut command: **swd**

Syntax

```
sscm showworkdirs [-mComputer] [-h] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>-m</i>		Computer to show working directories for. If <i>-m</i> is not specified then only working directories set for the current computer are displayed. If <i>-m</i> is specified without a computer name then all working directories for the given user will be returned.
-----------	--	---

workdir

Set the working directory.

Shortcut command: **wd**

Syntax

```
sscm workdir Directory Repository [-bBranch] [-h] [-r] [-yUsername:Password] [-zSCMServerAddr:PortNum]
```

<i>Directory</i>		Enter the full path to a local directory.
------------------	--	---

<i>Repository</i>	Enter the full path of the repository.
<i>-b</i>	Enter the branch name. If an option is not specified, the mainline branch is used.
<i>-r</i>	Lets the specified repository and its child repositories recursively inherit the working directory.

workdirinfo

Display branch and repository information for the working directory.

Shortcut command: **wdi**

Syntax

sscm workdirinfo *Directory* [-h] [-r]

<i>Directory</i>	Enter the full directory path.
<i>-r</i>	Recursively show information for all directories.

Chapter 4

CLI Quick Reference

This section includes an alphabetical list of each command and lists its alternative, if one is available. [Chapter 3, “Surround SCM CLI Commands,”](#) page 9 provides more detailed information about the commands.



Quick reference commands

Command	Definition
add	Add files to Surround SCM
addemailnotification aen	Add an email notification
addexistingmainline aeml	Add an existing mainline branch
addgroup ag	Add a security group
addtrigger at	Add a trigger
adduser au	Add a user
batch	Process batch commands
branchhistory bh	View branch history
branchproperty bp	Set and display branch security
breakshare bs	Unshare files
checkin ci	Check in files
checkout co	Check out files
diff	View differences between the file in your working directory and a version of the file or between two versions of the file
editemailnotification een	Edit an email notification
editgroup eg	Edit a security group
editmailqueue emq	Edit the status of pending messages in the server's mail queue

Command	Definition
editshadow es	Edit or update a shadow folder
edittrigger et	Edit a trigger
edituser eu	Edit a user
fetchttddb ftt	List the databases stored on the TestTrack Pro server
freeze	Lock a branch and do not allow changes to any files in the branch
get	Get files
history	View history
label	Create a label
ls	List all files and repositories in the repository
lsbranch lb	List all the branches in a repository and their corresponding parent branch
lsdefect ld	List the defects assigned to a TestTrack Pro user
lsemailnotification len	List all email notifications
lsgroup lg	Display security group information
lsmailqueue lmq	Display all mail in the queue
lsmainline lml	List all mainline branches
lsserverlog lsl	Display the server log
lsshadow lss	List shadow folders

Command	Definition
lstrigger lt	List all triggers
lstddb ltt	List the TestTrack Pro database configurations
lsuser lu	Display user information
merge	View differences between two files and merge changes into one file
mkbranch mb	Add a branch
mkmainline mml	Add a mainline branch
mkrepostiory mkrepo mr	Add a repository
mkshadow mks	Create a shadow folder
passwd pw	Set password
promote	Promote branch changes
promotefile pf	Promote file or repository changes
property	Set and display file or repository properties
rebase	Rebase branch changes
rebasefile rf	Rebase file or repository changes
rename	Rename a file or repository
renamebranch	Rename a branch
reporthistory rh	Generate a history report
restore	Restore a removed file or repository

Command	Definition
restorebranch	Restore a removed branch
retrieveuser rtu	Retrieve a global user from the Seapine License Server
rm	Remove a file or repository
rmbranch rb	Remove a branch
rmemailnotification ren	Remove an email notification
rmgroup rg	Remove a security group
rmmainline rml	Remove a mainline branch
rmserverlog rsl	Delete the server log
rmshadow rms	Remove a shadow folder
rmtrigger rt	Remove a trigger
rmtdb rtt	Delete a TestTrack database configuration
rmuser ru	Delete a user
rmworkdir rmwd	Remove a working directory
rollback rbk	Rollback a file to a previous version
securerepository sr	Set and display repository security
serveroption so	Set and display server options

Command	Definition
serveremailoption seo	Set server email options
setclient sc	Set and display client options
settdb stt	Add or edit a TestTrack database configuration
share	Link files
showworkdirs swd	Remove a working directory
uncheckout uco	Cancel a checkout command and lose any changes made to file(s)
unfreeze uf	Unlock a frozen branch
upgrademainline uml	Upgrade an existing mainline branch to the current database format
version	List the server and CLI versions
voidcheckout vco	Void a user's checkout operation
workdir wd	Set the working directory
workdirinfo wdi	Display branch and repository information for a working directory

Chapter 5

Surround SCM Glossary

This section provides definitions for terms used in Surround SCM.



Definitions

Ancestor branch

Branch a baseline, snapshot, or workspace branch is created from.

Baseline branch

Public branch that all users can access. Changes made to the baseline branch affect everyone who accesses that branch.

Branches

A branch uses an existing repository, and the files in that repository, as a starting point to enable concurrent development. When a repository is branched, the files in both branches are initially identical. As file contents change, the branched files become dissimilar. Use branches when you need to make changes to source files without affecting the existing repository.

Branch security

You can override the server security and enable/disable file commands for a specific branch in a repository. The security you set only affects one branch in the selected repository.

Diff

The diff command opens a utility that lets you quickly see if there are differences between server files and files in your working directory.

Email triggers

Triggers that use a standard email template, which can be customized, to inform users when an event occurs to a file or a set of files. Each email template is stored with the trigger on the server.

File types

You can add text, binary, and Mac binary files to Surround SCM.

Keywords

Case sensitive placeholders that can be inserted into text files and expanded upon check in. Keywords cannot be expanded for binary files. The following keywords are supported:

\$Author\$ - Name of the user who checked in the last revision

\$Date\$ - Date and time of the last revision

\$Header\$ - File, Revision, Date, Author

\$File\$ - Unqualified name of the file

\$Revision\$ - Revision number

\$Log\$ - File history in RCS-style format

Labels

Used to mark a specific version of a file or repository. Assigning labels allows for easy identification of all components with the same label.

Mainline branch

The highest-level branch that contains all source files, labels, other branches, and repositories. All files saved to a specific Surround SCM server are stored in a corresponding mainline branch.

Merge

The merge command opens a utility that lets you easily merge changes between server files and files in your working directory. You can merge changes into a new file.

Pre-event triggers

Triggers that are run after a client requests a command but before the event is complete. These trigger types let a server script perform additional checks on a file before the event completes. Pre-event triggers are for validation and custom text entry.

Post-event triggers

Triggers that run after a command is successfully completed on the server. These triggers can either run a script or send email. Post-event triggers are for logging and synchronization.

Triggers can only be fired from events on files, no triggers exist for branch-level or repository-level events. Files can be specified with a wildcard or a regular expression search so one trigger can affect a group of files. Each trigger is run once per file that it is associated with. Triggers can attach to the following file events:

Promote

Updates the ancestor branch with changes made in the selected branch, repository, or file.

Rebase

Updates the selected branch, repository, or file with changes made in the ancestor branch.

Recursive

If this option is selected, the command is applied to a repository and all files and subrepositories of that repository. For example, recursive check in simultaneously checks in all files and subrepositories of the selected repository.

Repository

A collection of files and/or subrepositories that helps you organize Surround SCM. Repositories are generally mapped to a directory on your hard drive.

Repository security

You can override the server security and enable/disable file commands for a selected repository. The security you set affects all branches in the selected repository.

Server security

Server security applies to all repositories in the mainline branch. Many companies will only set this type of security.

Shadow folder

Contains a “reference copy” of the current files in a branch. Files are automatically updated when changes are checked in to Surround. Shadow folders can provide read-only access to non-Surround users or provide a central location to build releases from.

Snapshot branch

A static branch of a baseline or workspace branch that generally corresponds to a project milestone, such as a QA build or final release build. Most Surround SCM commands are disabled in snapshot branches.

Timestamp

The recorded date and time of an event.

Triggers

Enhance and expand Surround by letting you run a script or send an email before, or after, a specific event. Triggers can be used for notifications, validation, custom text entry, logging, and synchronization.

Working directory

The local path where Surround SCM files are stored and changes are made to them.

Workspace branch

A private branch that is created by an individual user. Other users are not affected by changes made because the work is being done in a private branch. When a user finishes making changes, changes can be promoted from the workspace branch to the baseline, or ancestor, branch. Changes can also be rebased from the baseline branch to a workspace branch.

Index

A

About Surround SCM commands 10

About the guide 2

B

Branch commands 10

- addexistingmainline 11
- branchhistory 11
- branchproperty 11
- editshadow 12
- freeze 12
- lsbranch 13
- lsmainline 13
- lsshadow 13
- mkbranch 14
- mkmainline 14
- mkshadow 15
- promote 15
- rebase 16
- renamebranch 16
- restorebranch 17
- rmbranch 17
- rmmainline 18
- rmshadow 18
- unfreeze 18
- upgrademainlinebranch 19

C

Commands

- quick reference 62

Common command options 10

Configuring client options 6

Contacting Seapine support 3

F

File commands 19

- add 19
- batch 20
- breakshare 20

- checkin 21
- checkout 22
- diff 23
- get 23
- history 24
- label 25
- ls 25
- merge 26
- promotefile 27
- property 27
- rebasefile 28
- rename 29
- reporhistory 29
- restore 30
- rm 30
- rollback 31
- share 32
- undocheckout 32
- voidcheckout 33

G

Glossary

- ancestor branch 68
- baseline branch 68
- branch security 68
- branches 68
- diff 68
- email triggers 68
- file types 68
- keywords 68
- labels 69
- mainline branch 69
- merge 69
- post-event triggers 69
- pre-event triggers 69
- promote 69
- rebase 69
- recursive 69
- repository 69
- repository security 70
- server security 70
- shadow folder 70
- snapshot branch 70
- timestamp 70

- triggers 70
- working directory 70
- workspace branch 70

H

Help me! 3

K

Keywords 68

Q

Quick reference

- commands 62

R

Repository commands 33

- diff 33
- history 34
- label 34
- merge 35
- mkrepository 35
- promote file 36
- property 36
- rebase file 37
- rename 38
- restore 38
- rm 39
- secure repository 40

S

Security group commands 40

- addgroup 40
- editgroup 41
- ls group 41
- rm group 42

Server commands 42

- editmailqueue 42
- ls mailqueue 42

- ls serverlog 42
- rm serverlog 43
- serveremailoption 44
- serveroption 45
- setclient 46
- version 47

Starting Surround SCM 6

T

TestTrack Pro integration commands 47

- fetchtt db 47
- ls defect 48
- lsttdb 48
- rm ttdb 48
- set ttdb 48

Trigger commands 49

- addemailnotification 49
- addtrigger 50
- editemailnotification 52
- edittrigger 53
- ls emailnotification 54
- ls trigger 54
- rm emailnotification 55
- rm trigger 55

U

User commands 55

- adduser 55
- edituser 56
- ls user 57
- passwd 57
- retrieveuser 58
- rm user 58

W

Working directory commands 58

- showworkdirs 58
- workdir 58
- workdirinfo 59