Identity Router CLI Reference Guide

SecurID™

Contact Information

RSA Link at https://community.rsa.com contains a knowledgebase that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

RSA Conference Logo, RSA, and other trademarks, are trademarks of RSA Security LLC or its affiliates ("RSA"). For a list of RSA trademarks, go to https://www.rsa.com/en-us/company/rsa-trademarks. Other trademarks are trademarks of their respective owners.

License Agreement

This software and the associated documentation are proprietary and confidential to RSA Security LLC or its affiliates are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability.

This software is subject to change without notice and should not be construed as a commitment by RSA.

Third-Party Licenses

This product may include software developed by parties other than RSA. The text of the license agreements applicable to third-party software in this product may be viewed on the product documentation page on RSA Link. By using this product, a user of this product agrees to be fully bound by terms of the license agreements.

Note on Encryption Technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Use, copying, and distribution of any RSA Security LLC or its affiliates ("RSA") software described in this publication requires an applicable software license.

RSA believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." RSA MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

August 2021

Contents

Preface ... **5**
About This Guide ... 5
Audience ... 5
License Requirements ... 5
SecurID Support and Service ... 5
Support for SecurID Authentication Manager ... 5
Support for the Cloud Authentication Service and Identity Routers ... 5
RSA Ready Partner Program ... 6
Chapter 1: Installing the CLI ... **7**
Identity Router CLI ... 8
Installing the CLI on a UNIX Host ... 8
Installing the CLI on a Windows Host ... 9
Chapter 2: Syntax Conventions ... **12**
Syntax Conventions ... 13
Chapter 3: Common CLI Options ... **14**
Common CLI Options ... 15
Chapter 4: CLI Authentication ... **16**
CLI Authentication ... 17
Chapter 5: CLI Commands ... **18**
CLI Commands ... 19
idr-describe-applications ... 19
Syntax ... 19
Options ... 19
Example ... 20
idr-describe-keychains ... 20
Syntax ... 20
Options ... 20
Examples ... 21
idr-remove-keychains ... 21
Syntax ... 21
Options ... 22

Examples ... 22
idr-rename-keychain ... 22
Syntax ... 23
Options ... 23
Example ... 23
idr-replace-keychains ... 23
Syntax ... 24
Options ... 24
Example ... 24
idr-update-keychain-application ... 24
Syntax ... 25
Options ... 25
Example ... 25
idr-update-keychains ... 26
Syntax ... 26
Options ... 26
Example ... 26
idr-describe-user-profiles ... 27
Syntax ... 27
Options ... 27
Example ... 27
idr-remove-user-profiles ... 28
Syntax ... 28
Options ... 28
Example ... 29
Appendix A: Keychain File Formats ... 30
Keychain File Formats ... 31
Keychain CSV Format ... 31
Keychain CSV Input ... 31
Keychain XML Format ... 32
Sample XML Input ... 33

Preface

About This Guide

This guide provides instructions for using the command line interface to manage user profile data (application-specific credentials, also known as keychains) on the identity router.

Note: This guide does not apply to the embedded identity router that can be deployed on SecurID Authentication Manager.

Audience

This reference is intended for administrators and developers who need detailed information about the data types and operations supported by the identity router API.

License Requirements

This guide describes features that are available with the Enterprise and Premium licenses for SecurID. Contact your SecurID representative if you need to obtain a license.

For a complete list of SecurID documentation, see "SecurID Documentation" on RSA Link at https://community.rsa.com/docs/DOC-60094.

SecurID Support and Service

You can access community and support information on RSA Link at https://community.rsa.com. RSA Link contains a knowledgebase that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Support for SecurID Authentication Manager

Before you call Customer Support for help with the SecurID Authentication Manager appliance, have the following information available:

- Access to the SecurID Authentication Manager appliance.
- Your license serial number. To find this number, do one of the following:
  - Look at the order confirmation e-mail that you received when your ordered the product. This e-mail contains the license serial number.
  - Log on to the Security Console, and click **License Status**. Click **View Installed License**.
- The appliance software version. This information is located in the top, right corner of the Quick Setup, or you can log on to the Security Console and click **Software Version Information**.

Support for the Cloud Authentication Service and Identity Routers

If your company has deployed identity routers and uses the Cloud Authentication Service, SecurID provides you with a unique identifier called the Customer Support ID. This is required when you register with SecurID Customer Support. To see your Customer Support ID, sign in to the Cloud Administration Console and click **My Account > Company Settings**.

RSA Ready Partner Program

The RSA Ready Partner Program website at www.rsaready.com provides information about third-party hardware and software products that have been certified to work with SecurID products. The website includes Implementation Guides with step-by-step instructions and other information on how SecurID products work with third-party products.

Chapter 1: Installing the CLI

Identity Router CLI 8
Installing the CLI on a UNIX Host 8
Installing the CLI on a Windows Host 9

Identity Router CLI

The identity router command-line interface (CLI) is a set of terminal-based commands used to remotely manage the application-specific credentials (also known as keychains) stored on the identity routers in your deployment. You can install the CLI on any Windows or UNIX-based host with network access to the identity routers you want to manage. The SecurID Single Sign-On Web Services Software Development Kit (SWS SDK) includes the files required to install and configure the CLI.

Installing the CLI on a UNIX Host

Perform this procedure from the UNIX command console to install and configure the identity router CLI on a UNIX-based host.

Before you begin

- You must have administrative privileges on the host where you install the CLI.
- The host must be able to connect to the identity routers you want to manage using SSL and HTTPS protocols. For identity routers in the Amazon cloud, the host connects to the private IP address on port 9786. For on-premises identity routers, the host connects to the management interface IP address on port 443.
- Java 1.8 or later must be installed.

  Note: Oracle Java is required for UNIX-based hosts. The CLI does not work with Open JDK for UNIX.
- Obtain the identity router API Access ID and Access Key for your SecurID deployment. For instructions, see Enable Access to the Identity Router API in the *SecurID Cloud Administration Console Help*.
- Download **SWSSDK.zip** from RSA Link: https://community.rsa.com/docs/DOC-1051.

Procedure

1. Log on to the UNIX host using your administrator credentials.
2. Ensure that Java version 1.8 or later is installed:

   ```
   java -version
   ```
3. Extract the contents of **SWSSDK.zip** to a local directory.
4. Set the following environment variables:

   ```
   export SWS_SDK_HOME=</usr/local/SWSSDK>
   ```

   where *</usr/local/SWSSDK>* is the directory where you extracted the SDK files.

   ```
   export JAVA_HOME=</usr/local/jdk>
   ```

   where *</usr/local/jdk>*is the Java installation directory.

   ```
   export PATH=</usr/local/SWSSDK>/bin:$PATH
   ```

   where *</usr/local/SWSSDK>* is the directory where you extracted the SDK files.

```
export SWS_KEYFILE=</usr/local/SWSSDK>/key.txt
```

where </usr/local/SWSSDK> is the directory where you extracted the SDK files.

```
export SWS_SERVER=<xxx.xxx.xxx.xxx>
```

where <xxx.xxx.xxx.xxx> is the IP address for the identity router you want to manage. For identity routers in the Amazon cloud, use the private IP address. For on-premises identity routers, use the management interface IP address.

```
export SWS_PORT=<xxxx>
```

where <xxxx> is the port number for the connection. For identity routers in the Amazon cloud, use port 9786. For on-premises identity routers, use port 443.

Note: Setting the SWS_SERVER and SWS_PORT variables is optional. Setting the variables makes CLI commands easier to use because you do not need to add the -s=<xxx.xxx.xxx.xxx> and -p <portnumber> parameters to specify the port number and identity router IP address for every command. Set SWS_SERVER when managing only one identity router using the CLI. Do not set SWS_SERVER when managing multiple identity routers. Set SWS_PORT if all of your identity routers are on-premises or all are in the Amazon cloud. Do not set SWS_PORT if you have identity routers deployed in both environments.

5. Change the CLI commands to be executable:

   ```
   chmod +x $SWS_SDK_HOME/bin/*
   ```

6. Create a keyfile to specify the identity router API Access ID and Access Key:

   ```
   echo "key=<AccessID>/<AccessKey>" > </usr/local/SWSSDK>/key.txt
   ```

   where <AccessID> and <AccessKey> are the identity router API Access ID and Access Key, respectively, and </usr/local/SWSSDK> is the directory where you extracted the SDK files.

   Note: This keyfile contains sensitive security information. Use file permissions to restrict keyfile access to designated CLI users only.

7. Test the installation by requesting the version:
   ```
   idr --version
   ```

Results

After a successful installation, this command returns the version number of the CLI tool set, for example, 9.1. If unsuccessful, the command might return an error indicating a missing environment variable or a network communication problem.

Installing the CLI on a Windows Host

Perform this procedure from the Windows command prompt to install and configure the identity router CLI on a Windows host.

Before you begin

- You must have administrative privileges on the host where you install the CLI.
- The host must be able to connect to the identity routers you want to manage using SSL and HTTPS protocols. For identity routers in the Amazon cloud, the host connects to the private IP address on port 9786. For on-premises identity routers, the host connects to the management interface IP address on port 443.
- Java 1.8 or later must be installed.
- Obtain the identity router API Access ID and Access Key for your SecurID deployment. For instructions, see Enable Access to the Identity Router API in the *SecurID Cloud Administration Console Help*.
- Download **SWSSDK.zip** from RSA Link: https://community.rsa.com/docs/DOC-1051.

Procedure

1. Log on to the Windows host using your administrator credentials.
2. Check **Control Panel > Programs and Features** to ensure that Java version 1.8 (also listed as Java 8) or later is installed.
3. Extract the contents of **SWSSDK.zip** to a local directory.
4. Set the following environment variables:

   ```
   set SWS_SDK_HOME=<C:\SWSSDK>
   ```

   where <*C:\SWSSDK*> is the directory where you extracted the SDK files.

   ```
   set PATH=%PATH%;%SWS_SDK_HOME%\bin
   ```

   ```
   set SWS_KEYFILE=%SWS_SDK_HOME%\key.txt
   ```

   ```
   set JAVA_HOME=<C:\Program Files (x86)\Java\jre8>
   ```

   where <*C:\Program Files (x86)\Java\jre8*>is the Java installation directory.

   ```
   set SWS_SERVER=<xxx.xxx.xxx.xxx>
   ```

   where <*xxx.xxx.xxx.xxx*> is the IP address for the identity router you want to manage. For identity routers in the Amazon cloud, use the private IP address. For on-premises identity routers, use the management interface IP address.

   ```
   set SWS_PORT=<xxx.xxx.xxx.xxx>
   ```

   where <*xxxx*> is the port number for the connection. For identity routers in the Amazon cloud, use port 9786. For on-premises identity routers, use port 443.

   Note: Setting the SWS_SERVER and SWS_PORT variables is optional. Setting the variables makes CLI commands easier to use because you do not need to add the -s=<*xxx.xxx.xxx.xxx*> and -p <*portnumber*> parameters to specify the port number and identity router IP address for every command. Set SWS_SERVER when managing only one identity router using the CLI. Do not set SWS_SERVER when managing multiple identity routers. Set SWS_PORT if all of your identity routers are on-premises or all are in the Amazon cloud. Do not set SWS_PORT if you have identity routers deployed in both environments.

5. Create a keyfile to specify the identity router API Access ID and Access Key:

```
echo key=<AccessID>/<AccessKey>" > <c:\SWSSDK>\key.txt
```

where <*AccessID*> and <*AccessKey*> are the identity router API Access ID and Access Key, respectively, and <*c:\SWSSDK*> is the directory where you extracted the SDK files.

Note: This keyfile contains sensitive security information. Use file permissions to restrict keyfile access to designated CLI users only.

6. Test the installation by requesting the version:
   ```
   idr --version
   ```

Results

After a successful installation, this command returns the version number of the CLI tool set, for example, 9.1. If unsuccessful, the command might return an error indicating a missing environment variable or a network communication problem.

Chapter 2: Syntax Conventions

Syntax Conventions 13

Syntax Conventions

This table lists the conventions used to describe command-line syntax for identity router CLI commands, options, and input variables.

Convention	Description
`--debug` `-d`	Long and short form of a command option. • There is no difference in function. The long form may help clarify the meaning of the option. • Examples in this reference typically use the long form. • Use two dashes in front of the long form. • Use only one dash in front of the short form.
`[-h]`	Square brackets denote entire command options that are not required for a command to function.
<AccessID>	Pointed brackets indicate information that you must provide, such as a filename, Access ID, username, etc.
`-s | --server` *<ServerName | IPaddress>*	A vertical bar indicates a choice between two syntax items. The example at left indicates that you can choose between the short form (-s) and the long form (--server) of the option. You can also choose between specifying a server name and an IP address.

Note: All identity router CLI commands are in the **SWS_SDK_HOME\bin** directory. Windows commands end with .cmd. UNIX commands have no extension.

Chapter 3: Common CLI Options

Common CLI Options 15

Common CLI Options

This table describes the parameters common to all identity router CLI commands.

Option	Description	Required
`-d | --debug`	Displays debug information to help diagnose CLI command problems.	No
`-h | -help`	Displays help for the command.	No

`-k | --key <AccessID>/<AccessKey> | --keyfile <FileName>`	Specifies an authentication key. The key is a combination of the identity router API Access ID and Access Key configured for your SecurID deployment. Alternatively, you can specify a keyfile that contains the Access ID and Access Key. To specify a key, use one of the following:
	Short form (the forward slash is required):
	`-k <AccessID>/<AccessKey>`
	Long form (the forward slash is required):
	`--key <AccessID>/<AccessKey>`
	Specify a keyfile (long form required):
	`--keyfile <FileName>`
	The keyfile must contain the following line:
	`key=<AccessID>/<AccessKey>`
	For examples, see CLI Authentication on page 17
	* Not required if the SWS_KEYFILE environment variable is set to the keyfile. For instructions, see Identity Router CLI on page 8.
		Yes*

`-p | --port <PortNumber>`	Specifies the port number for the identity router API. **Not required if the SWS_PORT variable has been set to specify port 9786 for identity routers in the Amazon cloud or port 443 for on-premises identity routers.	Yes**
`-s | --server <ServerName | IPaddress>`	Specifies the identity router hostname or IP address. This option must point at a specific identity router, not a load balancer. *** Not required if the SWS_SERVER environment variable has been set to the identity router IP address. For instructions, see Identity Router CLI on page 8.	Yes***
`-v | --verbose`	Displays verbose output for the command.	No

Chapter 4: CLI Authentication

CLI Authentication 17

CLI Authentication

Each CLI command must specify an authentication key to gain access to the identity router API. This key is a combination of an administrator's identity router API Access ID and Access Key.

The following example shows how to specify an authentication key as a parameter for a CLI command:

```
idr-describe-keychains --server <myserver.myco.com> --key
<AccessID>/<AccessKey>
```

where <*AccessID*> and <*AccessKey*> are the identity router API credentials configured for a Super Admin account in the Cloud Administration Console.

Alternatively, you can use the --keyfile option to specify the Access ID and Access Key in a keyfile. The file must contain the following line:

```
key=<AccessID>/<AccessKey>
```

Note: This keyfile contains sensitive security information. Use file permissions to restrict keyfile access to designated CLI users only.

The following example shows how to specify a keyfile as a parameter for a CLI command:

```
idr-describe-keychains --server <myserver.myco.com> --keyfile
<~/.mysecretstuff/key.txt>
```

where <*~/.mysecretstuff/key.txt*> is the path and file name of the keyfile specifying the Access ID and Access Key.

Chapter 5: CLI Commands

CLI Commands 19
idr-describe-applications 19
idr-describe-keychains 20
idr-remove-keychains 21
idr-rename-keychain 22
idr-replace-keychains 23
idr-update-keychain-application 24
idr-update-keychains 26
idr-describe-user-profiles 27
idr-remove-user-profiles 28

CLI Commands

This chapter describes the commands included with the identity router CLI tool set, providing correct syntax, available options, and examples for each command. All identity router CLI commands are in the **SWS_SDK_HOME\bin** directory. All Windows commands end with the .cmd extension. UNIX commands do not have extensions.

Note: The clock difference between the CLI client host and the identity router may prevent the identity router from responding to client requests. If the client host is set to a time more than one minute before, or to any time ahead of the identity router clock, the CLI command fails.

idr-describe-applications

This command describes all keychain-related applications configured on the identity router. The command output specifies the application name, a list of credentials stored for each configured application, and whether users are able to edit their stored credentials for each application.

Syntax

```
idr-describe-applications -s | --server <ServerName> [-p <PortNumber>] -k <AccessID>/<AccessKey> | --keyfile <KeyFile> [-d] [-v]
```

Options

Option	Description	Required
`-d | --debug`	Displays debug information to help diagnose CLI command problems.	No
`-h | -help`	Displays command help.	No
`-k | --key <AccessID>/<AccessKey> | --keyfile <FileName>`	Specifies an authentication key, which is a combination of the identity router API Access ID and Access Key configured for your SecurID deployment, separated by a forward slash. Alternatively, you can use the --keyfile long form option, and specify a file that contains the key in the form: `key=<AccessID>/<AccessKey>` See Common CLI Options on page 15 for more information and CLI Authentication on page 17 for examples. * Not required if the SWS_KEYFILE environment variable is set to the keyfile.	Yes*
`-p | --port <PortNumber>`	Specifies the port number for the identity router API. **Not required if the SWS_PORT variable has been set to specify port 9786 for identity routers in the Amazon cloud or port 443 for on-premises identity routers.	Yes**
`-s | --server <ServerName | IPaddress>`	Specifies the identity router hostname or IP address. This option must point at a specific server, not a load balancer. ** *Not required if the SWS_SERVER environment variable has been set to the identity router IP address.	Yes***
`-v | --verbose`	Prints detailed command progress information.	No

Example

The following example displays the applications on identity router idr1.example.com using the Access ID abcdef and Access Key 123456:

```
idr-describe-applications --server idr1.example.com --key abcdef/123456
```

Application Name	Portal URL	Portal Text	Enable Keychain Edit	Credentials
SFDC	http://sfdc.com	SFDC	true	username1, password1
GOOGLE	http://google.com/apps	Google Apps	true	username2, password2

idr-describe-keychains

This command describes keychains for applications associated with the identity router. The command output specifies the application name and a list of credentials.

Syntax

```
idr-describe-keychains –s <ServerName> [-p <PortNumber>] –k
<AccessID>/<AccessKey> | --keyfile <KeyFile> [-a <ApplicationName>
[,<ApplicationName2>, ...] [-f <xml | csv | txt | names>] [-h] [-d] [-v]
```

Options

Option	Description	Required
`-a| --application <ApplicationName>`	Specifies the application for which keychains will be described. If you do not specify an application, the output lists keychains for all applications.	No
`-d | --debug`	Displays debug information to help diagnose CLI command problems.	No
`-f | --format <xml | csv | txt | names>`	Output format. Options include: XML, CSV, TXT (default) and NAMES. NAMES output lists only the usernames for the keychains.	No
`-h | -help`	Displays command help.	No
`-k | --key <AccessID>/<AccessKey> | --keyfile <FileName>`	Specifies an authentication key, which is a combination of the identity router API Access ID and Access Key configured for your SecurID deployment, separated by a forward slash. Alternatively, you can use the --keyfile long form option, and specify a file that contains the key in the form: `key=<AccessID>/<AccessKey>` See Common CLI Options on page 15 for more information and CLI Authentication on page 17 for examples. * Not required if the SWS_KEYFILE environment variable is set to the keyfile.	Yes*
`-p | --port <PortNumber>`	Specifies the port number for the identity router API. **Not required if the SWS_PORT variable has been set to specify port 9786 for identity routers in the Amazon cloud or port 443 for on-premises identity routers.	Yes**
`-s | --server`	Specifies the identity router hostname or IP address.	Yes***

Option	Description	Required
<*ServerName* | *IPaddress*>	This option must point at a specific server, not a load balancer. ** *Not required if the SWS_SERVER environment variable has been set to the identity router IP address.	
-v | --verbose	Prints detailed command progress information.	No

Examples

This example displays the keychains using the server and key options with the default outputs:

```
idr-describe-keychains --server idr1.example.com --key abcdef/123456
```

Username	Application	Credentials
user1	SFDC	username1, password1
user1	GOOGLE	username2, password2
user2	SFDC	username3, password3

This example displays the keychain for a specific user and produces output in XML format.

```
idr-describe-keychains --server idr1.example.com --key abcdef/123456 --
user user3 --format xml
<keychains>
<keychain>
<username>user3</username>
<application>
<name>SFDC</name>
<credential>
<name>username</name>
<value>********</value>
</credential>
<credential>
<name>password</name>
<value>********</value>
</credential>
</application>
</keychain>
</keychains>
```

idr-remove-keychains

This command removes keychains associated with users, or removes certain applications from user keychains.

Note: This command does not remove the associated user profile. See idr-remove-user-profiles on page 28 for more information.

Syntax

```
idr-remove-keychains –s <ServerName> [-p <PortNumber>] –k
<AccessID>/<AccessKey> | --keyfile <KeyFile> -u <UserName>[,<UserName2>,
...] [-a <ApplicationName>] [-h] [-d] [-v]
```

Options

Option	Description	Required
`-a | --application <ApplicationName>`	Specifies the application for which to remove keychains for users specified with the -u option.	No
`-d | --debug`	Displays debug information to help diagnose CLI command problems.	No
`-h | -help`	Displays command help.	No
`-k | --key <AccessID>/<AccessKey> | --keyfile <FileName>`	Specifies an authentication key, which is a combination of the identity router API Access ID and Access Key configured for your SecurID deployment, separated by a forward slash. Alternatively, you can use the --keyfile long form option, and specify a file that contains the key in the form: `key=<AccessID>/<AccessKey>` See Common CLI Options on page 15 for more information and CLI Authentication on page 17 for examples. * Not required if the SWS_KEYFILE environment variable is set to the keyfile.	Yes*
`-p | --port <PortNumber>`	Specifies the port number for the identity router API. **Not required if the SWS_PORT variable has been set to specify port 9786 for identity routers in the Amazon cloud or port 443 for on-premises identity routers.	Yes**
`-s | --server <ServerName | IPaddress>`	Specifies the identity router hostname or IP address. This option must point at a specific server, not a load balancer. *** Not required if the SWS_SERVER environment variable has been set to the identity router IP address.	Yes***
`-u | --username <UserName>`	Each instance of this parameter specifies the username of one user whose keychain data will be removed. You can add multiple instances to specify multiple users.	Yes
`-v | --verbose`	Prints detailed command progress information.	No

Examples

The following examples assume that the -k and -s values are defined by the SWS_SERVER and SWS_KEYFILE environment variables.

The following example removes the keychains associated with two users:

```
idr-remove-keychains --username tsmith --username jjohnson
```

The following example removes keychains for SampleApplication from two users:

```
idr-remove-keychains --username tsmith --username jjohnson --application
SampleApplication
```

idr-rename-keychain

This command transfers all keychain information (login credentials) from an existing user to a new user.

Note: This command fails if you try to rename the keychain to an existing user. Using **idr-rename-keychain** or **idr-remove-keychains** does not remove the user profile for the keychain. See idr-remove-user-profiles on page 28 for more information.

Syntax

```
idr-rename-keychain -s <ServerName> [-p <PortNumber>] -k
<AccessID>/<AccessKey> | --keyfile <KeyFile> -u <UserName> -n
<NewUserName> [-h] [-d] [-v]
```

Options

Option	Description	Required
`-d | --debug`	Displays debug information to help diagnose CLI command problems.	No
`-h | -help`	Displays command help.	No
`-k | --key <AccessID>/<AccessKey> | --keyfile <FileName>`	Specifies an authentication key, which is a combination of the identity router API Access ID and Access Key configured for your SecurID deployment, separated by a forward slash. Alternatively, you can use the --keyfile long form option, and specify a file that contains the key in the form: `key=<AccessID>/<AccessKey>` See Common CLI Options on page 15 for more information and CLI Authentication on page 17 for examples. * Not required if the SWS_KEYFILE environment variable is set to the keyfile.	Yes*
`-n | --newuser <NewUserName>`	New username to which the renamed keychain information will be assigned.	Yes
`-p | --port <PortNumber>`	Specifies the port number for the identity router API. **Not required if the SWS_PORT variable has been set to specify port 9786 for identity routers in the Amazon cloud or port 443 for on-premises identity routers.	Yes**
`-s | --server <ServerName | IPaddress>`	Specifies the identity router hostname or IP address. This option must point at a specific server, not a load balancer. *** Not required if the SWS_SERVER environment variable has been set to the identity router IP address.	Yes***
`-u | --username <UserName>`	Existing username whose keychain information will be transferred.	Yes
`-v | --verbose`	Prints detailed command progress information.	No

Example

The following example renames the keychain for user johndoe to user johnathandoe on server 1.2.30.1 using the keyfile located at jdoe/mykeyfile.txt:

```
idr-rename-keychain --server 1.2.30.1 --keyfile jdoe/mykeyfile.txt --
username johndoe --newuser johnathandoe
```

idr-replace-keychains

This command replaces all keychain information on an identity router. The input file must be in CSV or XML

format.

Note: Verify that all desired keychains are in the replacement file. For example, if the keychain being replaced has five entries but the new keychain file contains only two entries, the three extra entries in the original keychain are lost.

Syntax

```
idr-replace-keychains -s <ServerName> [-p <PortNumber>] -k
<AccessID>/<AccessKey> | --keyfile <KeyFile> -f <FileName> [-h] [-d] [-v]
```

Options

Option	Description	Required
`-d | --debug`	Displays debug information to help diagnose CLI command problems.	No
`-f | --filename <FileName>`	Specifies the name of an XML or CSV file that contains keychain information.	Yes
`-h | -help`	Displays command help.	No
`-k | --key] <AccessID>/ <AccessKey> | --keyfile <FileName>`	Specifies an authentication key, which is a combination of the identity router API Access ID and Access Key configured for your SecurID deployment, separated by a forward slash. Alternatively, you can use the --keyfile long form option, and specify a file that contains the key in the form: `key=<AccessID>/<AccessKey>` See Common CLI Options on page 15 for more information and CLI Authentication on page 17 for examples. * Not required if the SWS_KEYFILE environment variable is set to the keyfile.	Yes*
`-p | --port <PortNumber>`	Specifies the port number for the identity router API. **Not required if the SWS_PORT variable has been set to specify port 9786 for identity routers in the Amazon cloud or port 443 for on-premises identity routers.	Yes**
`-s | --server <ServerName | IPaddress>`	Specifies the identity router hostname or IP address. This option must point at a specific server, not a load balancer. *** Not required if the SWS_SERVER environment variable has been set to the identity router IP address.	Yes***
`-v | --verbose`	Prints detailed command progress information.	No

Example

The following example replaces the keychains on server 10.202.4.50 with the keychain information specified in the file newkeychains, using the authentication information in mykeyfile.txt:

```
idr-replace-keychains --keyfile mykeyfile.txt -f newkeychains -s
10.202.4.50
```

idr-update-keychain-application

This command updates keychain information for an application, and can update multiple user credentials for a single application, based on one or more -c <*CredentialName=CredentialValue*> pairs. Each credential must be

passed in as an individual <*CredentialName=CredentialValue*> pair. This command does not accept comma-delimited lists.

Syntax

```
idr-update-keychain-application -s <ServerName> [-p <PortNumber>] -k
<AccessID>/<AccessKey> | --keyfile <KeyFile> -u <UserName> -a
<ApplicationName> -c <CredentialName=CredentialValue> [-h] [-v] [-d]
```

Options

Option	Description	Required
`-a | --application <ApplicationName>`	Specifies the name of the application to which the keychain update applies.	Yes
`-c | --credential <CredentialName=CredentialValue>`	Specifies a name=value pair for an application credential you want to modify. You can add multiple instances of this command option to modify multiple credentials with a single command.	Yes
`-d | --debug`	Displays debug information to help diagnose CLI command problems.	No
`-h | -help`	Displays command help.	No
`-k | --key <AccessID>/<AccessKey> | --keyfile <FileName>`	Specifies an authentication key, which is a combination of the identity router API Access ID and Access Key configured for your SecurID deployment, separated by a forward slash. Alternatively, you can use the --keyfile long form option, and specify a file that contains the key in the form: `key=<AccessID>/<AccessKey>` See Common CLI Options on page 15 for more information and CLI Authentication on page 17 for examples. * Not required if the SWS_KEYFILE environment variable is set to the keyfile.	Yes*
`-p | --port <PortNumber>`	Specifies the port number for the identity router API. **Not required if the SWS_PORT variable has been set to specify port 9786 for identity routers in the Amazon cloud or port 443 for on-premises identity routers.	Yes**
`-s | --server <ServerName | IPaddress>`	Specifies the identity router hostname or IP address. This option must point at a specific server, not a load balancer. *** Not required if the SWS_SERVER environment variable has been set to the identity router IP address.	Yes***
`-u |--username <UserName>`	Specifies a single-sign-on username for the user whose application credentials you want to modify.	Yes
`-v | --verbose`	Prints detailed command progress information.	No

Example

The following example updates keychains for the application MyApp, specifying user1 as the new user name and

specifying mypassword as the new password:

```
idr-update-keychain-application -a MyApp -c username=user1 -c
password=mypassword
```

idr-update-keychains

This command updates keychain information for multiple users specified in a CSV or XML file.

Syntax

```
idr-update-keychains –s <ServerName> [-p <PortNumber>] –k
<AccessID>/<AccessKey> | --keyfile <KeyFile> -f <FileName> [-h] [-d] [-v]
```

Options

Option	Description	Required
`-d | --debug`	Displays debug information to help diagnose CLI command problems.	No
`-f | --filename <FileName>`	Specifies the XML or CSV file that contains keychain information.	Yes
`-h | -help`	Displays command help.	No
`-k | --key <AccessID>/<AccessKey> | --keyfile <FileName>`	Specifies an authentication key, which is a combination of the identity router API Access ID and Access Key configured for your SecurID deployment, separated by a forward slash. Alternatively, you can use the --keyfile long form option, and specify a file that contains the key in the form: `key=<AccessID>/<AccessKey>` See Common CLI Options on page 15 for more information and CLI Authentication on page 17 for examples. * Not required if the SWS_KEYFILE environment variable is set to the keyfile.	Yes*
`-p | --port <PortNumber>`	Specifies the port number for the identity router API. **Not required if the SWS_PORT variable has been set to specify port 9786 for identity routers in the Amazon cloud or port 443 for on-premises identity routers.	Yes**
`-s | --server <ServerName | IPaddress>`	Specifies the identity router hostname or IP address. This option must point at a specific server, not a load balancer. *** Not required if the SWS_SERVER environment variable has been set to the identity router IP address.	Yes***
`-v | --verbose`	Prints detailed command progress information.	No

Example

The following example updates keychain information from the file ~/.mysecretstuff/keychains.csv, and assumes that the -k and -s values are defined by the SWS_SERVER and SWS_KEYFILE environment variables:

```
idr-update-keychains –f ~/.mysecretstuff/keychains.csv
```

For examples of CSV and XML formats for input files, see Keychain File Formats on page 31.

idr-describe-user-profiles

This command describes all user profiles configured on the identity router. The command output displays either a list or a count of user profiles.

Syntax

```
idr-describe-user-profiles -s <ServerName> [-p <PortNumber>] -k
<AccessID>/<AccessKey> | --keyfile <KeyFile> -u <UserName>[,<UserName2>,
...] -f | [--format <xml|csv|txt>] [-h] [-d] [-v]
```

Options

Option	Description	Required
`-c | --count`	Returns the number of user profiles.	No
`-d | --debug`	Displays debug information to help diagnose CLI command problems.	No
`-f | --format <xml | csv | txt>`	Format of the output. Choose XML, CSV, or TXT (default).	No
`-h | -help`	Displays command help.	No
`-k | --key <AccessID>/<AccessKey> | --keyfile <FileName>`	Specifies an authentication key, which is a combination of the identity router API Access ID and Access Key configured for your SecurID deployment, separated by a forward slash. Alternatively, you can use the --keyfile long form option, and specify a file that contains the key in the form: `key=<AccessID>/<AccessKey>` See Common CLI Options on page 15 for more information and CLI Authentication on page 17 for examples. * Not required if the SWS_KEYFILE environment variable is set to the keyfile.	Yes*
`-p | --port <PortNumber>`	Specifies the port number for the identity router API. **Not required if the SWS_PORT variable has been set to specify port 9786 for identity routers in the Amazon cloud or port 443 for on-premises identity routers.	Yes**
`-s | --server <ServerName | IPaddress>`	Specifies the identity router hostname or IP address. This option must point at a specific server, not a load balancer. *** Not required if the SWS_SERVER environment variable has been set to the identity router IP address.	Yes***
`-u | --username <UserName> [,<UserName2> ...]`	One or more usernames of the users for whom the user profiles will be described.	Yes
`-v | --verbose`	Prints detailed command progress information.	No

Example

The following example shows how to display descriptions of the user profiles on the identity router at address 11.220.22.22 using the keyfile idr_cli_key.txt:

```
idr-describe-user-profiles -s 11.220.22.22 -keyfile idr_cli_key.txt
```

The command produces output in the following format (using the default text format):

```
Username
----------------
Administrator
indexer
indexeruser
myaccount
tester
----------------
```

The following example shows how to display the count of the user profiles on the server used in the previous example:

```
idr-describe-user-profiles -s 11.220.22.22 -keyfile idr_cli_key.txt -c
```

This command produces the following output:

```
5
```

idr-remove-user-profiles

This command completely removes one or more user profiles.

Syntax

```
idr-remove-user-profiles –s <ServerName> [-p <PortNumber>] –k
<AccessID>/<AccessKey> | --keyfile <KeyFile> -u <UserName>[,<UserName2>
...] [-h] [-d] [-v]
```

Options

Option	Description	Required
`-d | --debug`	Displays debug information to help diagnose CLI command problems.	No
`-h | -help`	Displays command help.	No
`-k | --key <AccessID>/<AccessKey> | --keyfile <FileName>`	Specifies an authentication key, which is a combination of the identity router API Access ID and Access Key configured for your SecurID deployment, separated by a forward slash. Alternatively, you can use the --keyfile long form option, and specify a file that contains the key in the form: `key=<AccessID>/<AccessKey>` See Common CLI Options on page 15 for more information and CLI Authentication on page 17 for examples. * Not required if the SWS_KEYFILE environment variable is set to the keyfile.	Yes*
`-p | --port <PortNumber>`	Specifies the port number for the identity router API. **Not required if the SWS_PORT variable has been set to specify port 9786 for identity routers in the Amazon cloud or port 443 for on-premises identity routers.	Yes**
`-s | --server <ServerName |`	Specifies the identity router hostname or IP address.	Yes***

Option	Description	Required
IPaddress>	This option must point at a specific server, not a load balancer. *** Not required if the SWS_SERVER environment variable has been set to the identity router IP address.	
`-u | --username <UserName>[, <UserName2>, <UserName3>, ...]`	Specifies the username of one or more user profiles that you want to remove. You can specify multiple usernames with a comma-separated list.	Yes
`-v | --verbose`	Prints detailed command progress information.	No

Example

The following example removes user profiles for the usernames testuser1 and testuser2 on identity router idr1.example.com, using Access ID abcdef and Access Key 123456:

```
idr-remove-user-profiles --server idr1.example.com --key abcdef/123456 -u
testuser1, testuser2
```

Appendix A: Keychain File Formats

Keychain File Formats 31
Keychain CSV Format 31
Keychain CSV Input 31
Keychain XML Format 32
Sample XML Input 33

Keychain File Formats

The **idr-update-keychains** and **idr-replace-keychains** commands have an option that specifies a file containing a set of keychain information. If the file extension is .csv, the file format is assumed to be a comma-separated-value (CSV) file. If the file extension is .xml, the file format is assumed to be XML.

Keychain CSV Format

The keychain CSV file format should follow the RFC 4180 (http://tools.ietf.org/html/rfc4180) proposal for CSV files. The CSV file format must have the following structure:

- The first line must be a header line with the columns named as follows:
    - The first column must be named "Username".
    - The second column must be named "Application Name".
    - Subsequent columns must be named in pairs for credential names and credential values. For example, the third column would be named "CredentialName1" and the fourth column named "CredentialValue1". The fifth column would be named "CredentialName2", and the sixth column named "CredentialValue2", and so on.
- Each subsequent line must contain the values for the user name, application name, and credential name/credential value pairs.
- To prevent a previously configured credential from being overridden, set the value of a credential to '********' (eight asterisks).
- If a user does not have an entry for a particular application, leave the values for the application credentials empty.

Keychain CSV Input

As a best practice, use the **idr-describe-keychains** command with the -f CSV option to return the existing keychain information from your identity router in CSV format, then copy and modify that output to serve as the input for future commands.

The following is an example of a CSV keychain file with three applications and two credentials:

```
Username,ApplicationName,CredentialName1,CredentialValue1,CredentialName2,
CredentialValue2
user1,SFDC,login,user1@myco.com,password,mypwd
user1,Google,username,user1,password,gmailpwd
user2,SFDC,login,user2,password,hispwd
```

The following is an example of a CSV keychain file with one application and four credentials:

```
Username,ApplicationName,CredentialName1,CredentialValue1,CredentialName2,
CredentialValue2,CredentialName3,CredentialValue3,CredentialName4,Credenti
alValue4
```

```
albert,,,,,,,,
Administrator,Concur,userid,andy,password,********,,,,
administrator,,,,,,,,
andrew,,,,,,,,
anthony,,,,,,,,
ashley,,,,,,,,
athos,,,,,,,,
dartagnon,,,,,,,,
porthos,,,,,,,,
maxwell,,,,,,,,
mycodevlab,,,,,,,,
```

Keychain XML Format

If the file extension is .xml, the file format is assumed to be XML that supports the XML schema specified in by the keychains-schema.xsd file in the SWS_SDK_HOME doc directory. This schema follows:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">


<xs:simpleType name="stringtype">

<xs:restriction base="xs:string"/>

</xs:simpleType>


<xs:complexType name="credentialtype">

<xs:sequence>

<xs:element name="name" type="stringtype"/>

<xs:element name="value" type="stringtype"/>

</xs:sequence>

</xs:complexType>


<xs:complexType name="applicationtype">

<xs:sequence>

<xs:element name="name" type="stringtype"/>

<xs:element name="credential" minOccurs="0" maxOccurs="unbounded"
type="credentialtype"/>

</xs:sequence>
```

```
</xs:complexType>

<xs:complexType name="keychaintype">
<xs:sequence>
<xs:element name="username" type="stringtype"/>
<xs:element name="application" minOccurs="0" maxOccurs="unbounded"
type="applicationtype"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="keychainstype">
<xs:sequence>
<xs:element name="keychain" minOccurs="0" maxOccurs="unbounded"
type="keychaintype"/>
</xs:sequence>
</xs:complexType>

<xs:element name="keychains" type="keychainstype"/>
</xs:schema>
```

Sample XML Input

As a best practice, use the **idr-describe-keychains** command with the -f XML option to return the existing keychain information from your identity router in XML format, then copy and modify that output to serve as the input for future commands.

The following is an example of an XML document with two users. User1 has credentials for two applications, and user2 has credentials for only one application:

```
<?xml version="1.0"?>
<keychains>
<keychain>
<username>user1</username>
<application>
<name>SFDC</name>
<credential>
```

```
<name>username</name>
<value>user1@myco.com</value>
</credential>
<credential>
<name>password</name>
<value>mypwd</value>
</credential>
</application>
<application>
<name>Google</name>
<credential>
<name>username</name>
<value>user1</value>
</credential>
<credential>
<name>password</name>
<value>gpassword</value>
</credential>
</application>
</keychain>
<keychain>
<username>user2</username>
<application>
<name>SFDC</name>
<credential>
<name>username</name>
<value>user2@myco.com</value>
</credential>
<credential>
<name>password</name>
```

```
<value>hispwd</value>
</credential>
</application>
</keychain>
</keychains>
```

The following is an example of an XML document with one application and four credentials. This example is the XML-format equivalent of the second example provided in Keychain CSV Format on page 31:

```
<?xml version="1.0"?>
<keychains>
<keychain>
<username>albert</username>
</keychain>
<keychain>
<username>Administrator</username>
<application>
<name>Concur</name>
<credential>
<name>userid</name>
<value>andy</value>
</credential>
<credential>
<name>password</name>
<value>********</value>
</credential>
</application>
</keychain>
<keychain>
<username>administrator</username>
</keychain>
<keychain>
<username>andrew</username>
```

```
</keychain>
<keychain>
<username>anthony</username>
</keychain>
<keychain>
<username>ashley</username>
</keychain>
<keychain>
<username>athos</username>
</keychain>
<keychain>
<username>dartagnon</username>
</keychain>
<keychain>
<username>porthos</username>
</keychain>
<keychain>
<username>mycodevlab</username>
</keychain>
</keychains>
```