



## Command Line Interface Guide

Version 3.1

© 2021 StrongBox Data Solutions, Inc.  
505 Maisonneuve West Suite 400  
Montreal, QC Canada H3A 3C2



© 2021 StrongBox Data Solutions, Inc. All rights reserved.

StrongBox Data Solutions, Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. StrongBox Data Solutions shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information which is protected by copyright. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of StrongBox Data Solutions. The information contained in this document is subject to change without notice.

Microsoft, Windows, Internet Explorer, and all Windows-based trademarks are either trademarks or registered trademarks of Microsoft Corporation.

Mozilla and Firefox are registered trademarks of the Mozilla Corporation.

Linux is a registered trademark of Linus Torvalds.

Copyright © 2009-2019 CentOS Licensed under the Creative Commons Attribution-Share Alike 3.0 License

<http://creativecommons.org/licenses/by-sa/3.0/legalcode>

Java and all Java-based trademarks are trademarks of Oracle.

IBM Enterprise Tape Library Driver © International Business Machines Corporation. All rights reserved.

Other product and company names may be the trademarks of their respective owners.

StrongBox Data Solutions, Inc.  
505 Maisonneuve West Suite 400  
Montreal, QC Canada H3A 3C2  
US Phone 512-349-0300  
EU Phone +49 7171 99800-11

# Contents

---

<b>StrongLink Command Line Interface</b>	<b>4</b>
Installing the StrongLink Command Line Interface .....	4
Logging into the StrongLink Command Line Interface .....	4
Listing Files and Namespaces .....	5
Retrieving Files .....	6
Archiving Files .....	6
Creating Queries .....	7
Listing Search Results .....	10
Retrieving Files Based on a Query .....	10
Assigning Metadata .....	11
Changing File Permissions .....	11
Moving Files and Namespaces .....	11
Renaming Files and Namespaces .....	11
Soft Deleting Files and Namespaces .....	11
Getting Help .....	11

# StrongLink Command Line Interface

---

The StrongLink command line interface (CLI) enables you to:

- List files in a specified path in the StrongLink global namespace.
- Retrieve files in a specified path in the StrongLink global namespace.
- Archive files in a directory to a specified path in the StrongLink global namespace.
- Create queries for files, and retrieve or list the files in the query results.
- Apply metadata to query results or files in a directory.
- Change file permissions (POSIX user, group, and mode).
- Move, rename, and soft deleted files.

## Installing the StrongLink Command Line Interface

The StrongLink Command Line Interface can be installed on any Linux client. The installer is provided as a .sh file, which should be installed as the root user (for example, `sudo ./slk-cli-tools.sh`). After installing the .sh file, add `/opt/stonglink/bin` to your PATH.

The StrongLink Command Line Interface requires Java JRE runtime version 11 or later.

The StrongLink Command Line Interface is compatible with both the Oracle and OpenJDK versions of the Java runtime.

## Logging into the StrongLink Command Line Interface

Before you can use the StrongLink CLI, you must log in with your domain credentials. When you log in, a security token is issued, which enables you to use the CLI for one month. After one month, you must log in again.

To log in to the CLI:

1. Enter `slk login`.
2. Enter the hostname or IP address of StrongLink.
3. Enter your username.
4. Enter your password.
5. Enter the StrongLink domain name. A **Login Successful** message appears to indicate you are logged in.

If you enter your credentials incorrectly, a **Login Unsuccessful - Incorrect Credentials** message appears, and you must start from step 1 again. If you enter an invalid hostname or IP address, a **Host not reachable** message appears.

**NOTE:** If your LDAP groups change, you must log in again for this to be reflected in the StrongLink CLI.

## Listing Files and Namespaces

You can list files and namespaces that are in a specific namespace (directory).

To list files and namespaces, enter `slk list <namespace path>`

For example, to list files and namespaces in the **/Archive/University** namespace, enter `slk list /Archive/University`.

For each file, the following information is displayed:

- Filename
- Path (in the global namespace)
- Size
- Mode - If the last character is **t**, the file is only stored on tape. Files on tape may take longer to retrieve, because the files must be copied from the tape to a filesystem store.
- User
- Group
- Modified

By default, POSIX owners and groups are displayed as names. If you want to display IDs, use the `-n` flag with the `slk list` command (`slk list -n <namespace path>`).

You can use an asterisk as a wildcard with this command. For example, `slk list /Archive/file*` can be used to list all files that start with "file" in the /Archive namespace.

By default, this command is not recursive. If you want to run it recursively, use the `-R` flag. When this flag is used, files are displayed by namespace.

When output to a terminal, this command paginates results, showing 25 at a time.

## Retrieving Files

You can retrieve files that are in a specific namespace (directory).

To retrieve files, enter `slk retrieve <namespace path or namespace path\filename> <local destination directory>`.

For example, to retrieve files in the **/Archive/University** namespace, enter `slk retrieve /Archive/University ./destination`.

When you retrieve files, the files are downloaded to your local computer. If the files are online (on NFS or SMB stores), the files are downloaded immediately. If files are offline, the files are copied to the Smart Pool bound to the namespace, and then downloaded (each file is downloaded as soon as it is copied to the bound Smart Pool).

You can use an asterisk as a wildcard with this command. For example, `slk retrieve /Archive/file* ./destination` can be used to download all files that start with "file" in the /Archive namespace to the destination directory.

By default, this command is not recursive. If you want to run it recursively, use the `-R` flag.

If a file is retrieved with the same name as a file that exists in the destination directory, the file name is updated to include `DUPLICATE_FILENAME` and the StrongLink ID and version number of the file.

## Archiving Files

You can archive files that are in a specific directory on your local computer. When you archive files, the files are uploaded to a specific StrongLink namespace.

To archive files, enter `slk archive <directory or filename> <namespace path>`

For example to archive files in the **Working** directory, and store the files in the **/Archive/Working** namespace, enter `slk archive Working /Archive/Working`.

Specify the full path of the directory you are archiving if it is not within the working directory.

You can use an asterisk as a wildcard with this command. For example, `slk archive ./source/file* /Working` can be used to upload all files that start with "file" in the /source directory to the /Working namespace.

When you archive files, the files are copied to the store that is bound to the specified namespace.

By default, this command is not recursive. If you want to run it recursively, use the `-R` flag.

**NOTE:** You must have write permissions to the namespace to which you archive. If the namespace to which you are archiving does not yet exist, it will be created.

# Creating Queries

You can create queries to find files that meet specific criteria.

To create a query, enter `slk search`, then enter the StrongLink query. StrongLink queries use the StrongLink Query Language. After you enter this command, a query ID is generated, which you can use to retrieve the specific files for the query or to tag those files with metadata.

A query contains at least one expression. Each expression contains the following fields:

- The StrongLink database field you want to search (for example, `resources.name`)
- An operator (for example, greater than, equals, containing, etc.)
- The value you want the operator to evaluate (for example, `file.txt`)

For example, a query expression is "Modified date greater than 2020-10-10" where "Modified date" indicates the field you want to search, "greater than" is the operator, and "2020-10-10" is the value.

Field	Operator	Value
<code>resources.modified</code>	<code>\$gt</code>	<code>2020-10-10</code>

The field by which you query determines the format of the value you must provide. For example, querying by a file size requires the value to be numerical values, but for a date field, the value would use a date format.

If a query includes multiple expressions, each subsequent expression includes a logical evaluator (AND or OR). This determines how the query evaluates multiple expressions.

You can use the following fields in your queries:

Field	Description
<code>path</code>	The namespace path in which the file exists
<code>resources._id</code>	The ID of the resource (database record for the file)
<code>resources.birth_time</code>	The date and time the file was initially created
<code>resources.created</code>	The date and time the resource was created in StrongLink
<code>resources.group_sid</code>	The group SID of the file
<code>resources.mimetype</code>	The MIME type of the file
<code>resources.modified</code>	The date and time the resource was last modified in StrongLink
<code>resources.mtime</code>	The date and time a file was last modified
<code>resources.name</code>	The name of the file
<code>resources.owner_sid</code>	The owner SID of the file
<code>resources.parent_id</code>	The ID of the namespace in which the file is stored
<code>resources.posix_gid</code>	The POSIX GID for the file
<code>resources.posix_mode</code>	The POSIX mode for the file
<code>resources.posix_uid</code>	The POSIX UID for the file
<code>resources.rcr_mtime</code>	The date and time the file's contents were last modified

resources.size	The size of the file
resources.version	The version of the files
rcr.store_object_ids	The ID of the store that the file is stored in
<udm.field_id>	The name of a user-defined metadata schema, then a period, and the ID of the field in the schema

You can use the following operators in your queries:

Operator	Description
\$	Projects the first element in an array that matches the query.
\$all	Finds arrays that contain all elements specified in the query.
\$and	Joins query clauses with a logical AND; finds resources that match the conditions of both clauses.
\$bitsAllClear	Finds numeric or binary values in which a set of bit positions all have a value of 0.
\$bitsAllSet	Finds numeric or binary values in which a set of bit positions all have a value of 1.
\$bitsAnyClear	Finds numeric or binary values in which any bit from a set of bit positions has a value of 0.
\$bitsAnySet	Finds numeric or binary values in which any bit from a set of bit positions has a value of 1.
\$elemMatch	Projects the first element in an array that matches the specified \$elemMatch condition.
\$eq	Finds values that are equal to a specified value.
\$exists	Finds resources that have the specified field.
\$expr	Allows use of aggregation expressions in the query language.
\$geoIntersects	Selects geometries that intersect with a GeoJSON geometry.
\$geoWithin	Selects geometries in a bounding GeoJSON geometry.
\$gt	Finds values that are greater than a specified value.
\$gte	Finds values that are greater than or equal to a specified value.
\$in	Finds any of the values specified in an array.
\$jsonSchema	Validates documents against a specified JSON schema.
\$lt	Finds values that are less than a specified value.
\$lte	Finds values that are less than or equal to a specified value.
\$meta	Projects the document's score assigned during \$text operation.
\$mod	Performs a modulo operation on the value of a field and selects documents with a specified result.
\$ne	Finds all values that are not equal to a specified value.
\$near	Returns geospatial objects in proximity to a point. Requires a geospatial index.
\$nearSphere	Returns geospatial objects in proximity to a point on a sphere. Requires a geospatial index.

\$nin	Finds none of the values specified in an array.
\$nor	Joins query clauses with a logical NOR; finds resources that fail to match both clauses.
\$not	Finds resources that do not match the query expression.
\$or	Joins query clauses with a logical OR; finds resources that match the conditions of either clause.
\$regex	Selects documents where values match a specified regular expression.
\$size	Finds resources if an array field is a specified size.
\$slice	Limits the number of elements projected from an array. Supports skip and limit slices.
\$text	Performs text search.
\$type	Finds resources if a field is of the specified type.
\$where	Finds resources that satisfy a JavaScript expression.

The following are examples of commonly used queries.

Query	Purpose
'{"resources.name":"filename.txt"}'	Find files with a specific file name
'{"rcr.store_object_ids":{"\$all":[1]}}'	Find files in a specific store
'{"resources.size":{"\$gt": 1048576}}'	Find files greater than one megabyte (sizes are in bytes)
'{"path":{"\$gte":"/Archive/Sales"}}'	Find files in a specific namespace
'{"resources.mimetype":"image/jpeg"}'	Find files of a specific MIME type
'{"resources.posix_uid":999}'	Find files for a specific UID
'{"resources.posix_gid":999}'	Find files for a specific GID
'{"resources.mtime":{"\$gt":"2021-10-10"}}'	Find files modified since a specific date
'{"netcdf.Institute": "ABC"}'	Find files based on NetCDF file metadata. Replace "Institute" with the field you want to search. Supported fields include: ID, Title, File Type, File Description, File Version, Institute, Institute_ID, Project_ID, Realm, Conventions, Creator, Contributor, Exp_ID, License, and Creation Date.
'{"project.name":"hadron"}'	Find files based on user-defined metadata. The user-defined schema and field name are the field. For example, if querying by the name file in the Project schema, the field you use in your query is Project.name.
'{"\$and": [{"resources.name": "projectfile.txt"}, {"project.name": "2020"}]}'	Find files based on user-defined metadata and with a specific file name

The search command also can be used with the following flags:

- `-name` - Search by file name
- `-user` - Search by POSIX owner
- `-group` - Search by POSIX group

For example, to search for a file called **file.txt**, enter `slk search -name file.txt`

## Listing Search Results

You can list files that are found for a specified query ID.

To list files, enter `slk list <query ID>`

For example, to list files for query ID 999, enter `slk list 999`.

For each file, the following information is displayed:

- Filename
- Path (in the global namespace)
- Size
- Mode
- User
- Group
- Modified
- Status (Online or Offline) - If this is blank, the file is online. Online files can be retrieved immediately, and are stored on NFS or SMB stores. If this is **Offline**, the file is stored on archival storage. Offline files may take longer to retrieve, because the files must be copied from archival storage.

By default, POSIX owners and groups are displayed as names. If you want to display IDs, use the `-n` flag with the `slk list` command (`slk list -n <query ID>`).

## Retrieving Files Based on a Query

You can retrieve files found for a specified query ID, which are online. For example, if a query finds 200 files, you can retrieve only those 200 files.

To retrieve files based on a query, enter `slk retrieve <query ID>`

For example to retrieve files for query ID 999, enter `slk retrieve 999`.

When you retrieve files, the files are downloaded to your local computer. If the files are online (on NFS or SMB stores), the files are downloaded immediately. If files are offline, the files are copied to the Smart Pool bound to the namespace, and then downloaded.

**NOTE:** When retrieving files based on a query, global namespace hierarchical structure is not preserved. If two files are retrieved with the same name, the file name is updated to include `DUPLICATE_FILENAME` and the StrongLink ID and version number of the file.

## Assigning Metadata

You can assign metadata to files in a specific namespace, or files found for a query.

To assign metadata, enter `slk tag <query ID or namespace path> <schema name>.<field name>=<UDM>`

You can add metadata for multiple fields at once.

For example, if you want to assign Miguel to the Author field and English to the Language field in the document schema for all files in the /Reports namespace, enter `slk tag /Reports document.Author=Miguel document.Language=English`

If you are assigning metadata with a space in it, such as a first and last name, put quotation marks around the metadata. For example, `slk tag /Reports document.Author="Miguel Cahill"`

By default, this command is not recursive. If you want to run it recursively, use the -R flag.

## Changing File Permissions

You can change the POSIX user, group, and mode for a file, or files in a global namespace path.

To change the owner for a file, enter `slk owner <POSIX UID or user name> <resource ID or global namespace path>`

To change the group for a file, enter `slk group <POSIX GID or group name> <resource ID or global namespace path>`

To change the mode for a file, enter `slk chmod <POSIX mode as an integer or string> <resource ID or global namespace path>`

For these commands, add -R to recursively change the POSIX user, group, or mode for all files in a specified namespace. For example, `slk chmod 766 -R /namespace`

## Moving Files and Namespaces

You can move a file or namespace to a different namespace in the StrongLink global namespace.

To move a file or namespace, enter `slk move <namespace or file path> <new namespace path>`

For example, to move `/directory/file.txt` to `/newfolder`, enter `slk move /directory/file.txt /newfolder`

## Renaming Files and Namespaces

You can rename a file or namespace.

To rename a file or namespace, enter `slk rename <namespace or file path> <new namespace or file name>`

For example, if you want to rename `/directory/file.txt` to `newname.txt`, enter `slk rename /directory/file.txt newname.txt`

## Soft Deleting Files and Namespaces

You can soft delete a file or namespace. When you soft delete a file or namespace, it is hidden from other users in StrongLink, but it will still take up storage space until it is hard deleted within StrongLink.

To soft delete a file or namespace, enter `slk delete <namespace or file path>`

For example, to soft delete `\directory\file.txt`, enter `slk delete \directory\file.txt`

## Getting Help

To display a list of available commands, enter `slk help`

**FOR MORE INFO VISIT**

[strongboxdata.com/stronglink](https://strongboxdata.com/stronglink)

© 2021 StrongBox Data Solutions, Inc.  
505 Maisonneuve West Suite 400  
Montreal, QC Canada H3A 3C2

