

# IBM System Storage TS4500 Tape Library

---



## Command Line Interface

The IBM TS4500 Command Line Interface (CLI) program can be used to access the TS4500 library from a CLI. It is an addition to TS4500 management GUI commands. The TS4500 CLI can be downloaded from the web at the following URL:

<http://www-01.ibm.com/support/docview.wss?uid=ssg1S4001255>

The TS4500 CLI provides the ability to access library functions through a command line interface. All view commands create a comma-separated value (csv) style output. The output is directed to the command prompt window by default, but can also be redirected to a file. To redirect the output to a file, you can add > output\_filename to the end of the command.

In order to use the TS4500 Tape Library CLI, the following prerequisites apply:

- Java 6.0 or later must be installed.

Cartridge movement using the TS4500 CLI is generally performed based on the current cartridge location:

### 1. Moving to storage

- If currently in storage, then use `prestageDataCartridges` only if library contains a High Density frame
- If currently in drive, then use `moveFromDrive`
- If currently in I/O with Virtual I/O enabled, then use `assignDataCartridges`

### 2. Moving to drive

- If currently in storage, then use `moveToDrive`
- If currently in drive, then use `moveFromDrive`, followed by `moveToDrive`
- If currently in I/O with Virtual I/O enabled, then use `assignDataCartridges` followed by `moveToDrive`

### 3. Moving to I/O

- If currently in storage, then use `removeDataCartridges`
- If currently in drive, then use `moveFromDrive` followed by `removeDataCartridges`
- If currently in I/O with Virtual I/O enabled, then use `removeDataCartridges`

Some move commands allow you to specify which cartridge to move by either its 8-character volser or its current location in the library. By supporting moves by location, cartridges with an Unknown volser can be moved via the CLI.

Moves from a storage slot location are supported by:

- `assignDataCartridges`
- `bulkAssignDataCartridges`
- `bulkAssignDataCartridgesByLogicalLibrary`
- `destageDataCartridges`
- `moveToDrive`
- `prestageDataCartridges`
- `removeDataCartridges`

The format for storage locations F#,C#,R#,T# for a library with a High Density frame. The format for I/O locations is F#,R#. As shown, the fields must be separated by commas without spaces.

Example: For a removeDataCartridges input file in a High Density library, you could have the following:

F1,C2,R8,T0  
F2,C1,R25,T3  
F1,C1,R10,T1

or using volser:

For an assignDataCartridges input file to the logical library **testlib**:

AAA000L4, testlib  
ABC001L4, testlib  
ZZZ000L4, testlib

You can run the CLI from any directory. However, it is simplest to run it from the directory where the TS4500CLI.jar file is stored. Running it from that directory enables you to execute the CLI commands without entering the path name of the TS4500CLI.jar file. However, if you run the CLI from any other directory, you must include the entire path name of the TS4500CLI.jar file as a part of the command. Running the CLI from the directory where the TS4500CLI.jar file is stored, Your command looks like this:

**java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --[cli\_command]**

## Supported Actions

### - **assignDataCartridges**

Use the **assignDataCartridges** command to assign or reassign data cartridges to a logical library.

### - **batch**

Use the **batch** command to instruct the system to perform multiple commands that are specified in a file.

### - **bulkAssignDataCartridges**

Use the **bulkAssignDataCartridges** command to assign cartridges in bulk, rather than assigning cartridges one at a time. You can bulk-assign up to 100 cartridges to a common logical library.

### - **bulkAssignDataCartridgesByLL**

Use the **bulkAssignDataCartridgesByLL** command to assign, in bulk, cartridges from a common logical library source to multiple logical library destinations.

### - **codeUpdate**

Use the **codeUpdate** command to update the library firmware.

### - **createBEP**

Use the **createBEP** command to create a bar code encryption policy.

### - **createKeyLabelMapping**

Use the **createKeyLabelMapping** command to create a key label mapping.

### - **createVolserRanges**

Use the **createVolserRanges** command to create one or more new VOLSER ranges.

**- deleteBEP**

Use the **deleteBEP** command to delete a bar code encryption policy.

**- deleteKeyLabelMapping**

Use the **deleteKeyLabelMapping** command to delete a key label mapping.

**- deleteVolserRanges**

Use the **deleteVolserRanges** command to delete VOLSER ranges.

**- destageDataCartridges**

Use the **destageDataCartridges** command to move cartridges from their cartridge cache locations (as specified in a text file) to a high-density slot.

**- downloadLog**

Use the **downloadLog** command to download a .zip file with the logs from all of the node cards. This command is useful for troubleshooting.

**- downloadPropertiesFile**

Use the **downloadPropertiesFile** command to download the properties file (TSSC file) if exists in the card.

**- downloadSnapshot**

Use the **downloadSnapshot** command to download a specific snapshot from the library.

**- driveCodeUpdate**

Use the **driveCodeUpdate** command to update the firmware in a specified drive or specified list of drives.

**- editBEP**

Use the **editBEP** command to edit a bar code encryption policy.

**- editKeyLabelMapping**

Use the **editKeyLabelMapping** command to edit a key label mapping.

**- getVIOStatus**

Use the **getVIOStatus** command to shows the status of the virtual I/O (VIO) flag on the library.

**- modifyAdvancedEncSettings**

Use the **modifyAdvancedEncSettings** command to make the advanced encryption settings more or less restrictive for data that is stored in a library.

**- modifyFibreChannelSettings**

Use the **modifyFibreChannelSettings** command to set ports speed and topology for a specified drive. Speed and topology are the same for both ports in the drive.

**- modifyVolserRanges**

Use the **modifyVolserRanges** command to modify existing VOLSER ranges.

**- modifyVolserReporting**

Use the **modifyVolserReporting** command to set whether the VOLSER is reported to the host in a six-character format or an eight-character format.

**- moveFromAllDrives**

Use the **moveFromAllDrives** command to eject all cartridges from all full drives

**- moveFromDrive**

Use the **moveFromDrive** command to eject a single cartridge from a drive and move it to the home storage location.

**- moveToDrive**

Use the **moveToDrive** command to move a cartridge from a storage location to a specified drive location.

**- prestageDataCartridges**

Use the **prestageDataCartridges** command to move cartridges from the high-density slots to cartridge cache locations specified in a text file.

**- removeDataCartridges**

Use the **removeDataCartridges** command to remove the data cartridges from a storage location to the I/O station.

**- resetDrive**

Use the **resetDrive** command to power cycle a specific drive.

**- resetNodeCards**

Use the **resetNodeCards** command to reset one or more specified node cards.

**- restoreConfiguration**

Use the **restoreConfiguration** command to restore the configuration database to the library.

**- saveConfiguration**

Use the **saveConfiguration** command save the configuration database to the user computer.

**- setAutoEjectCleaningCarts**

Use the **setAutoEjectCleaningCarts** command to set the enabled or disabled flag for the auto eject expired cleaning cartridges function in the library.

**- setDrivePortsId**

Use the **setDrivePortsId** command to set the ports IDs for a specified drive (Port 1 and Port 2).

**- setMaximumVIOCartridges**

Use the **setMaximumVIOCartridges** command to update the number of virtual I/O slots for the logical library.

**- setNMADetection**

Use the **setNMADetection** command to sets (to on or off) the NMA detection flag in the library.

**- setRolePermissions**

Use the **setRolePermissions** command with the output from the **viewRolePermissions** command. This command instructs the system to change the permissions for a specified role.

**- setScannerSpeed**

Use the **setScannerSpeed** command to set the speed of the scanner.

**- setSlotOffline**

Set offline a given slot

**- setSlotOnline**

Set online a given slot

**- setSSL**

Use the **setSSL** command to set the SSL flag to enabled or disabled.

**- showQueuedExports**

Sets flag to Show/Hide (true, false) queued exports

**- viewAccessor**

Use the **viewAccessor** command to show the accessor's status and usage statistics.

**- viewBEP**

Use the **viewBEP** command to show a list of all VOLSER ranges showing the BEP settings.

**- viewCleaningCartridges**

Use the **viewCleaningCartridges** command to view high-level information for all cleaning cartridges in a library.

**- viewDataCartridges**

Use the **viewDataCartridges** command to view high-level information for all data cartridges in a library.

**- viewDriveDetails**

The **viewDriveDetails** provides a display of several aspects of the drive that are useful for troubleshooting.

**- viewDriveSummary**

Use the **viewDriveSummary** command to show all of the drives in the library.

**- viewDriveVPD**

Use the **viewDriveVPD** command to show vital product data (VPD) information for all drives.

**- viewFibreChannel**

Use the **viewFibreChannel** command to show the Fibre Channel settings for both ports in each drive.

**. viewIoStation**

Use the **viewIoStation** command to view a summary of information for all cartridges in the I/O station.

**- viewKeyLabelMapping**

Use the **viewKeyLabelMapping** command to view the list of available key labels.

**- viewLogicalLibraries**

Use the **viewLogicalLibraries** command to list the high-level information for all logical libraries.

**- viewLogicalLibraryDetails**

Use the **viewLogicalLibraryDetails** command to show the name, type, maximum cartridges, the number of virtual I/O slot addresses, and the number of drive addresses in a specified logical library.

**- viewNodeCards**

Use the **viewNodeCards** command to show a summary of information about the node cards.

**- viewOfflineComponents**

Shows all slots currently offline

**- viewRolePermissions**

Use the **viewRolePermissions** command to display a list of all the permissions for a specified role.

**- viewRoles**

Use the **viewRoles** command to display a list of all the defined roles in the library.

**- viewSnapshots**

Use the **viewSnapshots** command to display a list of all available snapshots in the library.

**- viewSystemSummary**

Use the **viewSystemSummary** command to view the physical library system summary.

**- viewSystemSummaryDetails**

Use the **viewSystemSummaryDetails** command to view detailed information about a specified frame.

**- viewUsers**

Use the **viewUsers** command to display a list of all the users in the library.

**- viewVolserRanges**

Use the **viewVolserRanges** to list all of the VOLSER ranges in the physical library.

**- viewVolserRangesByLL**

Use the **viewVolserRangesByLL** to list all of the VOLSER ranges for each logical library in the physical library.

**Command Parameters**

The following CLI parameters are supported.

-? : Shows help

-- : -Action, REQUIRED

-ip : Tape library ip address, REQUIRED

-f : Specified frame #

-h : Shows help

-p : Tape Library Password

-r : Drive row #

--ssl: Use this flag when using a library with SSL enabled

-u : Tape Library User ID

-v or --verbose : The default mode for the CLI is “quiet”. In quiet mode, progress messages are not included in the output. When the -v or --verbose parameters is added, the progress messages are included in the output. When using the CLI with scripting, you probably will not want the progress messages included in the output

**SSL access:**

Like the TS4500 Tape Library Web Management GUI, the TS4500 CLI can be used with or without SSL enabled on the TS4500 Tape Library.

When accessing the library with SSL enabled you need to add the “ssl” flag to the list of parameters:

**java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --ssl --[cli\_command]**

## Command format:

### **assignDataCartridges**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --assignDataCartridges  
fileName.txt
```

Assigns data cartridges to a logical library. This command receives as input, a text file with one or more assign operations. The input file would take the following format:

volser or location #1, Logical Library name  
volser or location #2, Logical Library name

For **example**, if you wanted to assign four cartridges to the logical library named testltolib, you could create a text file named volsers.txt containing the following lines:

```
ZZZ000L4, testltolib  
ZZZ001L4, testltolib  
F1,C2,R8,T0, testltolib  
ZZZ002L4, testltolib
```

### **batch**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] -- batch fileName.txt
```

Perform actions specified within a file. Each line in the file will be interpreted as a new command. The input file would take the following format:

--commandName [commandParameters]

### **Example:**

fileName.txt will contain:

```
--viewLogicalLibraries  
--viewloStation  
--viewAccessor  
--viewRoles  
--viewLogicalLibraryDetails VIO255  
--bulkAssignDataCartridges bulk.txt,LTO_6
```

### **bulkAssignDataCartridges**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] -- bulkAssignDataCartridges  
fileName.txt, LibTest
```

Assigns multiple cartridges at the same time instead of one at a time. The cartridges will be assigned in bulks of size 100 as maximum. The bulk will be assigned to a same logical library.

### **Example:**

fileName.txt will contain:

F1,C2,R8,T0  
F2,C2,R8,T0  
F3,C2,R8,T0  
F1,C3,R18,T5

Or

A00023JA  
A00024JA  
A00025JA

### **bulkAssignDataCartridgesByLL**

*java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --bulkAssignDataCartridgesByLL fileName.txt*

Assigns multiple cartridges at the same time instead of one at a time. This command allows to assign data cartridges to different logical libraries (different destiny LL). The cartridges will be assigned in bulks of size 100 as maximum.

The input file would take the following format:

volser or location #1, Logical Library name  
volser or location #2, Logical Library name

#### **Example:**

fileName.txt will contain:

F1,C2,R8,T0, testltolib1  
F2,C2,R8,T0, testltolib2  
F3,C2,R8,T0, testltolib3

Or

A00034JA, testltolib1  
A00035JA, testltolib2  
A00036JA, testltolib3

### **codeUpdate**

*java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --codeUpdate [firmware image file]*

Updates library firmware. The firmware image file must have extension .afwz

### **createBEP**

*java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --createBEP [Volser start],[Volser end],[Logical library name],[media type], [KLM 1 index ], [KLM 2 index - optional]*

Creates a cartridges encryption policy



The parameters required for this command are:

Volser Start,

Volser End,

Logical library name

Media type : Possible values are LTO, 3592

Key label mapping index 1 : You need to run “viewKeyLabelMapping” command first, to get the current list of valid indexes.

Key label mapping index 2 : This is an optional value (used for JAG drives).

### **Example:**

This will create an encryption policy for the volser range TTT200-TTT300 with the logical library “test1”, media type 3592, and the KLM 1

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --createBEP  
TTT200,TTT300,test1,3592,1
```

### **createKeyLabelMapping**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --createKeyLabelMapping  
[keyLabelFrom],[Key mode],[keyLabelTo]
```

Create a key label mapping

The required parameters for this command are:

keyLabelFrom: This is a string that cannot be repeated in the list of key labels.

KeyModeFrom: The possible values are:

- Wrapped-Hash
- Wrapped-Default
- Wrapped-Clear
- Direct-Default-Set
- Direct-Specific

When using WRAPPED-DEFAULT keyLabelTo is disabled.

KeyLabelTo: This is a string.

The max length of keyLabelFrom and keyLabelTo is 50 characters.

### **Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --createKeyLabelMapping  
START,Wrapped-Hash,END
```

### **createVolserRanges**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --createVolserRanges  
fileName.txt
```

Creates new volser ranges. This command receives as input, a text file with one or more volser ranges. The input file would take the following format:

startRange,endRange,LLName,mediaType[LTO,3592],[TRUE/FALSE] (flag to enable this range only for new cartridges)

**Example:**

fileName.txt will contain:

UAA9RH,UBB9RH,Lib1,3592,FALSE

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --createVolserRanges  
fileName.txt
```

**deleteBEP**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --deleteBEP [BEP index]
```

Deletes a cartridge encryption policy. You need to execute “viewBEP” command before, to get the index of the BEP that will be deleted.

**deleteKeyLabelMapping**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --deleteKeyLabelMapping [KLM  
index]
```

Deletes a key label mapping. You need to execute “viewKeyLabelMapping” command before, to get the index of the key label mapping that will be deleted.

**deleteVolserRanges**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --deleteVolserRanges  
fileName.txt
```

Deletes all the volser ranges with the file. The input file would take the following format:

Logical Library Name,startRange,endRange

**Example:**

fileName.txt will contain:

Lib1,TUU9RH,TXX9RH

**destageDataCartridges**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --destageDataCartridges [textFile.txt]
```

or

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --destageDataCartridges [volser1],  
[volser2],...
```

Destages cartridges based on a list of locations (or volsers) in a file or a list of volsers in the command line. The cartridges must be in TIER = 0. The input file would take the following format:

F1,C2,R4,T0

F2,C5,R2,T0

F1,C4,R8,T0

F1,C2,R6,T0

or

A00045JA

A00046JA  
A00047JA

### **Example:**

Using a text file:

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --destageDataCartridges moves.txt
```

or

using a list of volsers:

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --destageDataCartridges  
A00000JA,A00001JA,A00002JA
```

### **downloadLog**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --downloadLog
```

Downloads the zip file with all the library logs.

### **downloadPropertiesFile**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --downloadPropertiesFile
```

Downloads the properties file (TSSC file) if exists in the LCC card.

### **downloadSnapshot**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --downloadSnapshot [Index]
```

Downloads the selected snapshot. This command must be used after “viewSnapshots” to get the index for the snapshot.

### **driveCodeUpdate**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --driveCodeUpdate [image file] -l [list  
of drives] --reset [reset option]
```

Updates drive firmware. This command requires the following parameters:

Filename of Fw image and list of drives to be updated. If the file is not in the same directory where CLI is running you need to include the whole path, i.e: C:\myPath\driveCode.afwz

The list of drives has this format:

F#C#R#,F#C#R#,... . The list must be preceded by “-l” (L in lowercase stands for Location)

There are “,” to separate drives, but not to separate frame,column and row

--reset : Possible values are : IMMEDIATE, UNLOAD, MANUAL. When using IMMEDIATE, the drives must be empty.

### **editBEP**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --editBEP [BEP index] --KLM1 [KLM1  
index] --BEP [true/false] --KLM2 [KLM2 index]
```

Edits a cartridges encryption policy. The parameters required for this command are:  
Index: You need to run “viewBEP” command first, to get the current list of valid indexes.  
Key label mapping index 1 : You need to run “viewKeyLabelMapping” command first, to get the current list of valid indexes. The parameter is --KLM1  
Key label mapping index 2 : This is an optional value. The parameter is --KLM2  
Bep enabled: Possible values are True , false. The parameter is --BEP

**Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --editBEP 1 --KLM1 3 --BEP false --KLM2 1
```

**editKeyLabelMapping**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --editKeyLabelMapping [KLM index], [Key label from],[Key mode],[Key label to]
```

Edit a key label mapping. You need to execute “viewKeyLabelMapping” command before, to get the index of the key label mapping that will be edited.  
The parameters required by this command are:

Index: You need to execute “viewKeyLabelMapping” to get this value.  
keyLabelFrom: This is a string that cannot be repeated in the list of key labels.  
KeyModeFrom: The possible values are:

- Wrapped-Hash
- Wrapped-Default
- Wrapped-Clear
- Direct-Default-Set
- Direct-Specific

When using WRAPPED-DEFAULT keyLabelTo is disabled.

KeyLabelTo: This is a string.  
The max length of keyLabelFrom and keyLabelTo is 50 characters.

**Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --editKeyLabelMapping 1,START,Wrapped-Hash,END2
```

**getVIOStatus**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --getVIOStatus
```

Shows the status of the VIO flag on the library.

**modifyAdvancedEncSettings**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --modifyAdvancedEncSettings LLName,Advanced Method [TRUE / FALSE], Advanced Policy[0-7], Density code[0-2], Key path[1,2,3,4,6]
```

Modifies the advanced encryption settings for a Logical Library. The advanced fields are:  
Advanced method, advanced policy, density code and key path.  
The parameters required by this command are:

Logical Library Name, Advanced Method [TRUE / FALSE], Advanced Policy\*, Density code\*\*, Key path\*\*\*

\*Advanced policy can take any of these possible values:

- No advanced setting = 0
- Don't encrypt if no policy = 1
- Encrypt if no policy = 2
- Policy required = 3
- Never encrypt (policy override) = 4
- Always encrypt (policy override) = 5
- Internal label – Selective Encryption = 6
- Internal label – Encrypt all = 7

\*\*Density code can take any of these possible values:

- No advanced setting = 0
- Shows Encryption = 1
- Masks Encryption = 2

\*\*\*Key path can take any of these possible values:

- No advanced setting = 1
- System = 2
- Application (IBM) = 3
- Application (T10) = 4
- Library = 6

### Example:

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --modifyAdvancedEncSettings  
Library1,TRUE,2,2,2
```

### modifyFibreChannelSettings

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --modifyFibreChannelSettings  
[speed,topology]-f [Drive frame] -c [Drive column] -r [Drive row]
```

Sets ports speed and topology for the given drive. Speed and topology will be the same for both ports.

The parameters required by this command are:

[Speed ,Topology] -f [frame] -c [column] -r [row]

For Speed valid values are: Auto,1,2,4,8. The values are in Gb/s.

For Topology valid values are: Auto-L,L,N,Auto-N

### Example:

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --modifyFibreChannelSettings 4,N -f  
1 -c 1 -r 1
```

### **modifyVolserRanges**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --modifyVolserRanges  
fileName.txt
```

Allows to modify Logical Library, media Type and the flag to enable the volser range only for new cartridges.

The parameters required by this command are:

FileName.

Each line in the file will be interpreted as a volser range. The parameters are:

startRange\*,endRange\*,LLName,mediaType [LTO,3592],[TRUE/FALSE] (flag to enable only for new cartridges).

\* : These values are used to get the range index in the table, so those values cannot be modified.

The input file would take the following format:

starRange,endRange,LL name, media Type [LTO,3592], Enable [true/false].

### **Example:**

fileName.txt will contain:

UAA9RH,UBB9RH,Library 1,3592,TRUE

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --modifyVolserRanges fileName.txt
```

### **modifyVolserReporting**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --modifyVolserReporting [Logical  
library name],[6/8]
```

Modifies Volser Reporting value to host (6 or 8) .

### **Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --modifyVolserReporting Lib1,6
```

### **moveFromAllDrives**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --moveFromAllDrives
```

Demounts all full drives.

### **moveFromDrive**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --moveFromDrive -f [frame index] -c  
[column index] -r [row index]
```

*Ejects a cartridge from a given drive.*

**Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --moveFromDrive -f 1 -c 4 -r 1
```

**moveToDrive**

Using cartridge location:

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [user] -p [password] --moveToDrive F#,C#,R#,T# -f [Drive frame index] -c [Drive column index] -r [Drive row index]
```

or

using cartirdge volser:

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [user] -p [password] --moveToDrive [Volser] -f [Drive frame index] -c [Drive column index] -r [Drive row index]
```

Mounts a cartridge to the given drive.

**prestageDataCartridges**

Using a text file:

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [user] -p [password] --prestageDataCartridges [text file name]
```

or

using a list of volsers

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [user] -p [password] --prestageDataCartridges [Volser 1],[Volser 2],...
```

Prestages cartridges based on a list of locations (volsers) in a file. Cartridges must be in TIER >= 3.

The input file would take the following format:

F#,C#,R#,T#

F#,C#,R#,T#

F#,C#,R#,T#

or

[VOLSER1]

[VOLSER2]

Example:

F1,C2,R4,T3

F2,C5,R2,T3

F1,C4,R8,T4

F1,C2,R6,T4

VOLSER is also accepted

A00045JA

A00046JA

A00047JA

Using a text file:

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [user] -p [password] --prestageDataCartridges moves.txt
```

or

using a list of volsers:

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [user] -p [password] --prestageDataCartridges  
A00000JA,A00001JA,A00002JA
```

### **removeDataCartridges**

Using a text file

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --removeDataCartridges  
fileName.txt
```

or

using a list of volsers

```
>> java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --removeDataCartridges  
[Volser 1],[Volser 2],...
```

Remove the data cartridges (cartridges moved to the I/O station).

The input file would take the following format:

volser or location #1

volser or location #2

### **Example:**

fileName.txt will contain:

F1,C2,R8,T0

F1,C3,R18,T5

Or

A00045JA

A00047JA

Using a text file:

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --removeDataCartridges  
fileName.txt
```

or

using a list of volsers:

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --removeDataCartridges  
A00000JA,A00001JA,A00002JA
```

### **resetDrive**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --resetDrive -f [Drive frame index] -c  
[Drive column index] -r [Drive row index]
```

Power cycles a given drive

### **resetNodeCards**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --resetNodeCards [Cards list]
```



Resets all specified node cards

The required parameters are:

A comma separated list of the nodes to be reset. Accepted values are:

ALL,ALLLCA,ALLACC,ALLSMC,ALLXYC,ALLPLUS,XYCPLUS,ACCPLUS,LCAPLUS,XYA,XYB,OPC1,  
SMC[2-16],ACCA,ACCB,LCA[1-31],LCAB[1-31]

### **restoreConfiguration**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --restoreConfiguration [configuration backup file .db]
```

Restores CONFIGURATION database to the library.

The required parameter is:

File name: If the file is not in the same directory where CLI is running you must include the full path.

#### **Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --restoreConfiguration configuration_backup.db
```

### **saveConfiguration**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --saveConfiguration
```

Saves CONFIGURATION database to the user computer. The file will be save in the same directory where the CLI is running. The file name is "configuration\_backup.db".

### **setAutoEjectCleaningCarts**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --setAutoEjectCleaningCarts [ENABLED/DISABLED]
```

Sets ENABLED/DISABLED the auto eject expired cleaning cartridges flag in the library.

The required parameter is:

flag value:

ENABLED for ON

DISABLED for OFF

#### **Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --setAutoEjectCleaningCarts ENABLED
```

### **setDrivePortsId**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --setDrivePortsID [port 1 id],[port 2 id] -f [Drive frame index] -c [Drive column index] -r [Drive row index]
```

Sets the ports Ids for a given drive (Port 1 and Port 2).

#### **Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --setDrivePortsID 18,21 -f 1 -c 1 -r 1
```

### **setMaximumVIOCartridges**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --setMaximumVIOCartridges  
Lib1,200
```

Update the number of Virtual I/O slots for the logical library.

The required parameters are:

The lower limit of the accepted values is the number of physical IO slots. If the library has two IO magazines, each with 18, the valid range of values will be : [36-255]

Logical library name, value [Total number of IO slots – 255]

#### **Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --setMaximumVIOCartridges  
Lib1,200
```

### **setNMADetection**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --setNMADetection [TRUE/FALSE]
```

Sets ON/OFF the NMA detection flag in the library.

### **setRolePermissions**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --setRolePermissions [text file with  
accesses] -role [role name]
```

This command is used with “viewRolePermissions” output. The user will be able to change the existing permissions for a given role using the “viewRolePermissions” output saved in a text file. The possible permissions are:

“Read only”: User will have only read access to the web page.

“Modify”: User can execute edit operation in the web page.

“No Access”: User won't be able to see the web page.

The output file from “viewRolepermissions” can be edited to change any of the access levels for the listed pages, using the above values (Read only, modify, no access).

This new file will be used as a parameter for “setRolePermissions” command. The new values will be set for the given role.

#### **Example:**

The file can contain all of the pages or just the ones that will change.

| <b>Action,</b>                 | <b>Access Level</b> |
|--------------------------------|---------------------|
| Cartridges,                    | No Access           |
| Cartridges by Logical Library, | No Access           |
| Cleanning Cartridges,          | No Access           |

|                            |           |
|----------------------------|-----------|
| Drives,                    | No Access |
| Drives by Logical Library, | No Access |
| Email Notifications,       | Read Only |
| Email Recipients,          | Read Only |
| Encryption Internal,       | Read Only |
| Encryption Key Manager,    | Read Only |

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --setRolePermissions
rolePermissions.txt -role Service
```

### **setScannerSpeed**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --setScannerSpeed [Speed value]
```

Sets scanner speed.

The possible values for speed are:

speed: 0,30,40,50,60,70,80,90,100

0 = Nominal speed

### **Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --setScannerSpeed 100
```

### **setSlotOffline**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [user] -p [password] --setSlotOffline -f [Frame index] -c
[Column index] -r [Row index]
```

Set offline a given storage slot.

The required parameters are:

-f Frame index  
-c Column index  
-r Row index

when using r = 0 , all the rows for the given column will be set offline

### **Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [user] -p [password] --setSlotOffline -f 1 -c 3 -r 1
```

### **setSlotOnline**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [user] -p [password] --setSlotOnline -f [Frame index] -c
[Column index] -r [Row index]
```

Set online a given storage slot.

The required parameters are:

- f Frame index
- c Column index
- r Row index

when using  $r = 0$  , all the rows for the given column will be set offline

#### **Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [user] -p [password] --setSlotOnline -f 1 -c 3 -r 1
```

#### **setSSL**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --setSSL [ENABLED/DISABLED]
```

Enables or disables connection using Secure Socket Layer

#### **showQueuedExports**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --showQueuedExports [Logical library name],[TRUE/FALSE]
```

Set flag to Show/Hide (true, false) queued exports

The required parameters are:

Logical library name, Flag value [True/false] : SHOW = TRUE, HIDE = FALSE

#### **Example:**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --showQueuedExports Lib1,TRUE
```

#### **viewAccessor**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewAccessor
```

Shows accessor's status and usage statistics.

#### **viewBEP**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewBEP
```

Show a list of all the volser ranges showing the BEP settings.

### **viewCleaningCartridges**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewCleaningCartridges
```

Shows all the cleaning cartridges in the library

### **viewDataCartridges**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewDataCartridges
```

Shows all the data cartridges in the library

### **viewDriveDetails**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewDriveDetails -f [Drive frame index] -c [Drive column index] -r [Drive row index]
```

Shows details of the given drive.

### **viewDriveSummary**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewDriveSummary
```

Shows a list with some details of all the drives in the library

### **viewDriveVPD**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewDriveVPD
```

Shows VPD information for all drives

### **viewFibreChannel**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewFibreChannel
```

Shows the Fibre Channel settings for both Ports in each drive.

### **viewIoStation**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewIoStation
```

Shows all the cartridges in IO station.

## **viewKeyLabelMapping**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewKeyLabelMapping
```

Shows the list of available key labels.

## **viewLogicalLibraries**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewLogicalLibraries
```

Lists all the logical libraries. Queued Exports, VOLSER Reporting (6/8 characters) are CLI only columns.

## **viewLogicalLibraryDetails**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewLogicalLibraryDetails [Logical Library Name]
```

Shows the details of the given logical library.

## **viewNodeCards**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewNodeCards
```

Shows information on the node cards

## **viewOfflineComponents**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [user] -p [password] --viewOfflineComponents
```

View offline storage slots

## **viewRolePermissions**

Saving output to a text file:

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewRolePermissions [roleName] > rolePermissions.txt
```

Displaying output in command window

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewRolePermissions [roleName]
```

Displays a list of all the permissions for a given role. The output can be display in the command window or save to a text file using ">". When user save the output the file can be use in "setRolePermissions" command. It will show a list of the web pages available in the library and its corresponding access level.

## **viewRoles**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewRoles
```

Displays a list of all the defined roles in the library

## **viewSnapshots**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewSnapshots
```

*Shows the list of available snapshots in the library.*

## **viewSystemSummary**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewSystemSummary
```

Shows the physical library system summary. It will display information about each frame in ascending order, and at the end a summary with total numbers on the whole library will be displayed.

## **viewSystemSummaryDetails**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewSystemSummaryDetails  
[Frame number]
```

Shows physical library system summary for a particular frame

## **viewUsers**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewUsers
```

Displays a list of all the users in the library.

## **viewVolserRanges**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewVolserRanges
```

Shows all the volser ranges in the physical library.

## **viewVolserRangesByLL**

```
java -jar TS4500CLI.jar -ip [LCC ip] -u [username] -p [password] --viewVolserRangesByLL
```

Shows all the volser ranges assigned to each logical library.