Command specifications

ADD

Name

`add` - Add data set to the repository

Synopsis

`add [-v] <name> [<description>]`

Description

Add the specified data set with path `<name>` with optional `description` to the repository.

Options

`-v` verbose progress

Parameters

`<path>` The path of the data set.

`<name>` The name the data set will have in the repository

`<description>` **Optional** A description of the data set

Output

The meta data of the data set added, tab separated. Format:
<id> \t <name> \t <timestamp> \t <description> \t <number of files> \t <size>
In case of abort a detailed error message

Exit status

0 if successful,
1 if the file/folder under the given `<name>` does not exist
2 if the data set could not be added to the repository
4 if any other error occurred

Time Estimation

15 hours of work, in particular testing Java I/O with large data sets,testing various operating systems and privilege levels (admin vs non-admin), building methods for creating meta data and creating the structure of the application.
Since all of us have recent working experience in java, we add on top of that an estimated 2 hours per person to get our work flow set up.

REMOVE

Name

`remove` - remove a data set

Synopsis

`remove [-v] <name>`

Description

Remove the data set with given `<name>`

Options

`-v` verbose progress

Parameters

`<name>` The path of the data set (e.g. the name of the file/folder)

Output

No output, if the operation ended successfully. In case of abort e detailed error message.

Exit status

0 if successful,
1 if the data set with the specified `<name>` does not exist
2 if the data set could not be removed
4 if any other error occurred

Time Estimation

3 hours of work

COPY

Name

`copy` - copies a data set from the repository to any location in the file system.

Synopsis

`copy [-v] <name> <location>`

Description

Copies the data set specified with `<name>` to the given `<location>`. If the operation succeeded, the data set remains in the repository and is accessible with the original name at `<location>`.

Options

`-v` verbose progress

Parameters

`<id>` id of the data set.

`<location>` Path of the new location for the data set.

Output

No output, if the operation ended successfully. In case of abort e detailed error message.

Exit status

0 if successful,
1 if the data set with the specified `<id>` does not exist or the `<location>` parameter is invalid
2 if the data set could not be copied (e.g. if a file / folder with given name already exists at `<location>`)
4 if any other error occurred

Time Estimation

2 hours of work if done after the add and remove commands, since they are quite similar in syntax and copy is very similar to the add command.

REPLACE

Name

`replace` - replaces a data set with another one.

Synopsis

`replace [-v] <name> <path>`

Description

Replaces the data set `id` with the data set located at `<path>`.

Options

`-v` verbose progress

Parameters

`<id>` The id of the data set that should be replaced.

`<path>` The path to the new data set

Output

The new meta data of the data set replaced, tab separated. Format:
<id> \t <name> \t <timestamp> \t <description> \t <number of files> \t <size>
In case of abort a detailed error message

Exit status

0 if successful,
1 if command-parameter specific error occurred
2 if the data set with specified `<id>` does not exist or the `<path>` parameter is invalid
4 if any other error occurred

Time Estimation

2 hours of work. Very similar to the copy command

LIST

Name

`list` - lists all data set in the repository and their meta data

Synopsis

`list [-v]`

Description

Lists all data sets and their meta data.

Options

`-v` verbose progress

Output

A list of all data sets in the repository. Separated with tabs. Format:
<id> \t <name> \t <timestamp> \t <description> \t <number of files> \t <size> \n
<id> \t <name> \t <timestamp> \t <description> \t <number of files> \t <size>
Or a detailed error message in case of abort.

Exit status

0 if successful,
4 if any error occurred

Time Estimation

6 hours of work, mainly for traversing the metadata-storage and testing the behaviour of `<list>` with various commands

INIT

Name

`init` - initializes the repository in the current working directory

Synopsis

```
init [-v]
```

Description

Initializes the repository in the current working directory. During the initialization files for storing meta data are generated.

Options

`-v` verbose progress

Output

No output if the operation ended successfully. In case of abort a detailed error message.

Exit status

0 if successful,
4 if any error occurred

Time Estimation

3 hours of work since I/O testing on various operating systems and privilege levels should have already been done with the add command

SEARCH

Name

`search` - searches for one or more data sets

Synopsis

`search [-v][-a | -o] [-name=<pattern>] [-description=<pattern>] [-id=<id>] [-timestamp=<time>] [-size=<size>] [-files=<amount>]`

Description

Searches for data sets based on its meta data. Based on the parameters different meta data is considered.

Options

`-v` verbose progress

`-a` match all given search parameters

`-o` match one of the given search parameters

Parameters

`-name=<pattern>` The pattern the name should match

`-description=<pattern>` The pattern the description should match

`-id=<id>` ID of the data set

`-timestamp=<time>` Timestamp of the data set

`-size=<size>` The size in Bytes of the data set

`-files=<amount>` The amount of files the data set contains

output

A list of all data sets in the repository matching the search pattern, Separated with tabs. Format:
<id> \t <name> \t <timestamp> \t <description> \t <number of files> \t <size> \n
<id> \t <name> \t <timestamp> \t <description> \t <number of files> \t <size>
No output if none of the data sets matches the pattern. Or a detailed error message in case of abort.

Exit status

0 if successful,
1 if a command-parameter specific error occurred
2 if the command could not be executed
4 if any other error occurred

Time Estimation

6 hours of work, in particular for testing parameters