Moving or Renaming a Collection or File via the CLU

If your user account has the group administrator or system administrator role, and has the Write or Own permission level on an existing collection in DME (on all collections in the path), you can move one or more data files or collections to that existing collection in the following ways:

- If that existing collection has been configured to contain another collection, you can move a collection to it.
- If that existing collection has been configured to contain data files, you can move a file to it.

You can also rename a collection or file.

To move or rename one or more collections or files:

1. In your local system, create a JSON file that specifies a list of source and destination locations, as follows:

```
{
  "moveRequests": [
    {
      "sourcePath": "<full path in DME to the file/collection to be moved>",
      "destinationPath": "<full path in DME to the new file/collection>"
    },
    {
      "sourcePath": "<full path in DME to the file/collection to be moved>",
      "destinationPath": "<full path in DME to the new file/collection>"
    }
  ]
}
```

2. Make sure that each sourcePath and destinationPath pair in the JSON file meets the following criteria:
    - In the value of your destinationPath, the last part (collection name or data file name) is new.
    - In the value of your destinationPath, the parent collection already exists.
3. Run the following command:

```
dm_rename [optional parameters] <move-description.json>
```

The following table describes each parameter:

Parameter	Description
[-h]	If you want to print a usage (help) message for this command, specify this option.
[-D <REST-response>]	An optional parameter, specifying a path and filename in your local system. The system always creates a response file: • If you specify this parameter, the system saves the response from the server to the specified file in the specified location. • If you omit this parameter, the system saves the file as rename-response-header.tmp in your home directory.
[-o <rename-status-file>]	An optional parameter, specifying a path and filename in your local system. The system always creates a status file: • If you specify this parameter, the system saves the status to the specified file in the specified location. • If you omit this parameter, the system prints the status to standard output and saves the status as rename-response-message.json.tmp in your home directory. The status includes a result for each request (each source path and destination path pair). If the system was able to satisfy the request, the result is true. Otherwise, it's false. The status file provides an explanatory message in these cases. The status also includes an overall result. If all request statuses are true, then the overall result is true. Otherwise, it's false.
<move-description.json>	A path to the JSON file containing the source and destination locations.

For example, consider the following command:

```
dm_rename /NCI/JaneDoe/my-move-list.json
```

With the following code in the my-move-list.json file, the command renames a collection from Project_Orig to Project_New and then renames a file from File_Orig to File_New:

```
{
  "moveRequests": [
    {
      "sourcePath": "/Example_Archive/PI_Lab2/Project_Orig",
      "destinationPath": "/Example_Archive/PI_Lab2/Project_New"
    },
    {
      "sourcePath": "/Example_Archive/PI_Lab2/Project_2/File_Orig",
      "destinationPath": "/Example_Archive/PI_Lab2/Project_2/File_New"
    }
  ]
}
```

In this example, each sourcePath and destinationPath pair in the JSON file meets the above-mentioned criteria, as follows:

- At command runtime, the Project_New collection name and File_New file name do not already exist.
- At command runtime, the PI_Lab2 collection, Project_Orig collection, Project_2 collection, and File_Orig file already exist.
- In the first pair, only the Project_New collection name has changed. In the second pair, only the File_New file name has changed.