Updating Data File Metadata via the CLU

If your user account has the Write or Own permission level on an existing collection in DME, you can use a CLU command to update the metadata of a data file in that collection.

The character limit for each metadata value is 2700.

To update the metadata of a data file:

1. In your local system, create a JSON file that specifies the metadata for the data file:

```
{
    "metadataEntries": [
      {
        "attribute": "description",
        "value": "my-dataObject-description"
      },
      {
        "attribute": "example_date",
        "value": "20201231",
        "dateFormat": "yyyyMMdd"
      }
    ]
}
```

2. For each date attribute, specify one of the following date formats, and specify the date value in that format:

    - yyyyMMdd
    - yyyy.MM.dd
    - yyyy-MM-dd
    - yyyy/MM/dd
    - MM/dd/yyyy
    - MM-dd-yyyy
    - MM.dd.yyyy

    The system parses your date using the date format you specify. Then however, if the date attribute has a metadata validation rule in a different format, the system stores the date in the format specified by that rule.
3. In your JSON file, if you want to update the metadata of the parent collection, also specify the metadata for the parent collection. Click the following link to view the syntax:

```
{
    "metadataEntries": [
        {
         "attribute": "project",
         "value": "my-project-name"
        },
        {
         "attribute": "notes",
         "value": "my-project-notes"
        }
    ],
    "createParentCollections": true,
        "parentCollectionsBulkMetadataEntries": {
                "pathsMetadataEntries": [{
                        "path": "/Example_Archive/PI_Lab1/Project_New",
                        "pathMetadataEntries": [{
                                        "attribute": "collection_type",
                                        "value": "Folder"
                                },
                                {
                                        "attribute": "example info",
                                        "value": "123456"
                                }]
                }]
        }
}
```

4. Run the following command:

```
dm_register_dataobject [optional parameters] <description.json> <destination-path>
```

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 dataObject-registration-response-header.tmp in your home directory.
[-o <output-json-file>]	An optional parameter, specifying a path and filename in your local system. The system always creates an output file: • If you specify this parameter, the system saves the output to the specified file in the specified location. • If you omit this parameter, the system saves the output as dataObject-registration-response-message.json.tmp in your home directory. If the command is successful, the output file is empty.
`<description.json>`	A path within a local system, including the name of the JSON file that specifies the metadata for the data file you intend to update.
`<destination-path>`	A path within DME, including the name of the file you intend to update.

For example, the following command updates the metadata of the data.txt file in the Project1 collection in DME:

```
dm_register_dataobject /NCI/JaneDoe/my-metadata.json /Example_Archive/PI_Lab1/Project1/data.txt
```

The JSON file must contain metadata for the data file, data.txt.