

Registering Directory Contents from Your Local System via the CLU

If your user account has the Write or Own permission level on an existing collection in DME, and if that existing collection has been configured to contain another collection, you can recursively register data (copy files within folders) from your local system into that existing collection. By default, the `dm_register_directory` command registers all files and directories in the source directory.

The character limit for each metadata value is 2700.

Contents of This Page

- [Registering Directory Contents](#)
- [Verifying Uploaded Files](#)
- [Example](#)

Registering Directory Contents

1.
 - a. Make sure that the destination collection (where you want to register your data) already exists. For instructions, refer to the following pages:
 - [Creating an Empty Collection via the CLU](#)
 - [Creating an Empty Collection via the GUI](#)
 - [Registering Data in Bulk via the GUI](#)
 - b. Optional: To register a subset of the files in the source directory, prepare to use one of the following options:
 - You can create a flat list of data files. In each line of this file, specify a relative path to the source path. Plan to use the `-l` option in the command.
 - You can use regex patterns to exclude files from registration, include files, or both.
 - To include files in registration, create an include file and plan to use the `-i` option in the command.
 - To exclude files from registration, create an exclude file and plan to use the `-e` option in the command.

For information, refer to [Specifying Include Criteria](#). If you proceed with a command that specifies both include and exclude files, the system first identifies files that match any of the *include* regex patterns. Then the system removes from the registration job the files that match any of the *exclude* patterns.

- c. Specify the metadata required for the collections and data files you intend to create in DME. For each directory and each data file that you intend to register, create a metadata file with the name `<[directory,filename]>.metadata.json`. Create each metadata file in the same path as its corresponding directory or data file. For example, for a file named `sample.txt`, name the corresponding metadata file `sample.txt.metadata.json`. (Keep in mind that the system registers the metadata files along with the corresponding data, unless you specify otherwise with a flat list or regex patterns.) In each metadata file, specify the metadata that you want to include, as follows:

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

- d. 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.

- e. Optional: If you want to create a parent collection for the directory contents (or update the metadata of an existing parent collection), also specify in your JSON file the metadata for the parent collection. To view the syntax, click the following link:

```
{
  "metadataEntries": [
    {
      "attribute": "description",
      "value": "description"
    },
    {
      "attribute": "example_date",
      "value": "20201231",
      "dateFormat": "yyyyMMdd"
    }
  ],
  "createParentCollections": true,
  "parentCollectionsBulkMetadataEntries": {
    "pathsMetadataEntries": [{
      "path": "/Example_Archive/PI_Lab1/Project_New",
      "pathMetadataEntries": [{
        "attribute": "collection_type",
        "value": "Folder"
      },
      {
        "attribute": "example info",
        "value": "123456"
      }
    ]
  }
}]
}
```

2. Run the following command:

```
dm_register_directory [optional parameters] <source-path> <destination-path>
```

The following table describes each option:

Option	Description
-c	If you want to turn on checksum calculation for validation of data transfer, specify this option. By default, the system does not perform this calculation because it can increase the time required to register the data, depending on the file size. <ul style="list-style-type: none">If the collection is 50 MB or larger, the system saves the computed checksum as user metadata (source_checksum). This checksum metadata contains the eTag.If the collection is smaller than that threshold, the system uses the computed checksum to perform checksum verification upon uploading the file. The system also saves the checksum as system metadata (checksum).
-d	If you want to preview a list of the source file(s)/folder(s) that the system would register based on what you specified as the source path, specify this dry run option. (This option tests only the source path. It does not test the destination path.)
-e <path-to-exclude-file>	If you want to exclude files, specify this option. The system excludes the files that match the patterns specified in the file <path-to-exclude-file>.
-h	If you want to print a usage (help) message for this command, specify this option.
-i <path-to-include-file>	If you want to include files, specify this option. The system includes the files that match at least one of the patterns specified in <path-to-include-file>.
-l <files-list>	If you want to register specific files and exclude all others, specify this option. The system registers the files mentioned in the <files-list> only. In each line of the <files-list> file, specify a relative path to the <source-path>.
-m	If you want to register metadata only (for files that already exist), specify this option. The system does not register files.

-s	If you want to skip the default confirmation prompt and register directly, use this option.
-t <num-threads>	If you want to use this parameter, contact NCIDataVault@mail.nih.gov for guidance.
-x	If you want to extract metadata from the header of TIFF or BMP image files, use this option.

The following table describes each parameter:

Parameter	Description
<source-path>	A path in your local system, specifying the data that you want to register. The system registers the contents of the folder you specify.
<destination-path>	A path within DME, specifying where you want the system to create the data.

- If the system prompts you to confirm the registration, type Y and press Enter. The command recursively registers files and directories from the directory specified in the <source-path> to the <destination-path>.

Verifying Uploaded Files

To verify uploaded files, use the `dm_get_dataobject` command. The system-generated metadata `data_transfer_status` displays ARCHIVED for files successfully uploaded to DME. For details, refer to [Retrieving the Metadata of a Data File via the CLU](#).

Example

To view an example, click the following link:

The following command copies the *contents* (folders and files) of the JaneDoe folder in the local system to the Project_New collection in DME. (The command does not copy the JaneDoe folder itself.)

```
dm_register_directory -i include.txt -e exclude.txt /NCI/JaneDoe /Example_Archive/PI_Lab1/Project_New
```

The example command specifies the following include file:

```
**/images/**
**/data/**
```

All of the paths in the include file are relative to the source folder, as specified in the command. This file includes the following files:

- All files in an images folder in the source folder (and subfolders).
- All files in a data folder in the source folder (and subfolders).

The example command specifies the following exclude file:

```
**.metadata.json
**/log/**
*.properties
```

All of the paths in the exclude file are relative to the source folder, as specified in the command. This file excludes the following files:

- All files in the source folder (and subfolders) with filenames that end in `.metadata.json`.
- All files in any log folder in the source folder (and subfolders).
- All files in the source folder with filenames that end in `.properties`.