

Using dm_register_dataobject

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 data files, you can register a data file into that collection. If the file is larger than ten GB, register it from a Globus endpoint, an AWS S3 bucket, or Google Cloud rather than from your local system. (To update metadata, refer to [Updating Data File Metadata via the CLU](#).)

The character limit for each metadata value is 2700.

To register a data file:

1. Choose whether to register from your local system, a Globus endpoint, or an S3 bucket.
 - To register from your local system, plan to use the source file parameter to specify the file that you want to register.
 - To register from Globus, S3, or Google Cloud, plan to use a JSON file to specify the file that you want to register.
2. In your local system, create a JSON file that specifies the metadata for the new data file. The contents of this file depend on the source of your data:
 - If you are registering from your local system, specify the metadata that you want to include. Click the following link to view the syntax:

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

To extract metadata from the header of TIFF or BMP image files, include the following line:

```
"extractMetadata" : true,
```

- If you are registering from a Globus endpoint, specify that endpoint, the file path on that endpoint, and the metadata that you want to include. Click the following link to view the syntax:

```
{
  "globusUploadSource": {
    "sourceLocation": {
      "fileContainerId": "globus-shared-endpoint-uid",
      "fileId": "file-path-on-shared-globus-endpoint"
    }
  },
  "metadataEntries": [
    {
      "attribute": "description",
      "value": "my-file-description"
    },
    {
      "attribute": "example_date",
      "value": "20201231",
      "dateFormat": "yyyyMMdd"
    }
  ]
}
```

- If you are registering from AWS S3, specify the S3 bucket, path, access key, secret access key, region, and the metadata that you want to include. Click the following link to view the syntax:

```
{
  "s3UploadSource": {
    "sourceLocation": {
      "fileContainerId": "s3-bucket-name",
      "fileId": "s3-object-key"
    },
    "account": {
      "accessKey": "aws-access-key",
      "secretKey": "aws-secret-key",
      "region": "aws-region"
    }
  },
  "metadataEntries": [
    {
      "attribute": "description",
      "value": "my-file-description"
    },
    {
      "attribute": "example_date",
      "value": "20201231",
      "dateFormat": "yyyyMMdd"
    }
  ]
}
```

- If you are registering from Google Cloud storage, specify the storage location, access token, and the metadata that you want to include. (For instructions on generating the access token, refer to https://cloud.google.com/storage/docs/reference/libraries#setting_up_authentication.) Click the following link to view the syntax:

```
{
  "googleCloudStorageUploadSource": {
    "sourceLocation": {
      "fileContainerId": "dme-upload-bucket",
      "fileId": "api-docs_UAT.json"
    },
    "accessToken": "accessToken",
    "accessTokenType": "SERVICE_ACCOUNT"
  },
  "metadataEntries": [
    {
      "attribute": "description",
      "value": "my-file-description"
    },
    {
      "attribute": "example_date",
      "value": "20201231",
      "dateFormat": "yyyyMMdd"
    }
  ]
}
```

3. 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.

4. In your JSON file, if you want to create a parent collection for the data file you are registering (or update the metadata of a parent collection), also specify the metadata for the parent collection. Click the following link to view the syntax:

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

5. Run the following command:

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

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: <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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. <p>If the command is successful, the output file is empty.</p>
<description.json>	A path to the JSON file that specifies the metadata for the new data file.
<destination-path>	A path within DME, including the name of the file you intend to register. Specify where you want the system to create the new data file. (If you specify an existing data file, this command updates the metadata for that data file. For details, refer to Updating Data File Metadata via the CLU.)
[source-file]	A path to a file in your local system: <ul style="list-style-type: none"> If you are registering from your local system, use this parameter to specify the file that you want to register. If you are registering from Globus, S3, or Google Cloud, omit this parameter.

For example, the following command registers the data.txt file from the JaneDoe folder in the local system to the Project_New collection in DME:

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

If you registered the file from Globus, S3, or Google Cloud, you can view the progress of the registration in the GUI. For instructions, refer to [Viewing Registration Status](#).