

Creating an Empty Collection via the CLU

You can use a CLU command to create an empty collection in DME. With the same command, you can create a parent collection for the empty collection. This command has the following prerequisites for the existing collection in which you intend to create the empty collection or the parent collection of the empty collection:

- Your user account has the Write or Own permission level on the existing collection.
- That existing collection has been configured to contain another collection.

The character limit for each metadata value is 2700.

To create an empty collection:

1. In your local system, create a JSON file that specifies the metadata for the new collection. At a minimum, this file must specify the collection type and all required metadata, as follows:

```
{
  "metadataEntries": [
    {
      "attribute": "collection_type",
      "value": "Project"
    },
    {
      "attribute": "project_start_date",
      "value": "20201231",
      "dateFormat": "yyyyMMdd"
    }
  ]
}
```

If an attribute has predefined values, specify values that fall within that set of values. 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.

2. In your JSON file, if you want to create a parent collection for the empty collection (or update the metadata of an existing parent collection), also specify the metadata for the parent collection. Click the following link to view the syntax:

```
{
  "metadataEntries": [{
    "attribute": "collection_type",
    "value": "Project"
  },
    {
      "attribute": "project_start_date",
      "value": "20201231",
      "dateFormat": "yyyyMMdd"
    }
  ],
  "createParentCollections": true,
  "parentCollectionsBulkMetadataEntries": {
    "pathsMetadataEntries": [{
      "path": "/Example_Archive/PI_Lab1",
      "pathMetadataEntries": [{
        "attribute": "collection_type",
        "value": "PI_Lab"
      }],
      {
        "attribute": "data_owner",
        "value": "example owner"
      },
      {
        "attribute": "data_generator",
        "value": "example data generator"
      }
    ]
  }
}
```

3. Run the following command:

```
dm_register_collection [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: <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 collection-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 collection-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 collection.
<destination-path>	A path within DME, including the name of the collection you intend to create. Specify where you want the system to create the new collection. (If you specify an existing collection, this command updates the metadata for that collection. For details, refer to Updating Collection Metadata via the CLU .)

The command registers the metadata specified in <description.json> to the <destination-path> in DME.

For example, the following command creates a Project_New collection within the PI_Lab1 collection and applies to that new collection the metadata in the my-collection.json file:

```
dm_register_collection my-collection.json /Example_Archive/PI_Lab1/Project_New
```

The JSON file must contain metadata for the new collection, Project_New. If the parent collection, PI_Lab1, does not already exist, then the JSON file must also contain metadata for that collection.