

Rclone

Rclone is a tool for managing material in cloud services. In DDC, it is primarily a way of transferring content out of a donor's cloud storage into DDC's digital storage.

- [Setting up Rclone](#)
 - [General setup](#)
 - [OneDrive or Sharepoint](#)
 - [Google Drive](#)
- [Using Rclone](#)
 - [Appraisal or preparing for a transfer](#)
 - [Creating a list of files and basic metadata](#)
 - [Finding if a transfer contains Google drive formats](#)
 - [Copying files](#)
 - [Basic copying](#)
 - [Shared folder copying](#)
 - [Google drive copying](#)
 - [Extracting checksums](#)
 - [Confirming fixity](#)
- [Next steps and packaging files](#)

Setting up Rclone

Rclone should already be installed in BitCurator but if you are transferring content that you need to log in to access, you'll need to do some additional set up to connect to cloud storage service. This will have to be repeated for each cloud storage service and possibly reconnected if some time has passed between the last time you've used Rclone. Below lists the general set up process with sections for specific cloud storage providers used in practice so far. If there is a cloud storage provider you need to access that's not covered below, please contact the digital archivist.

General setup

This general setup will work with DropBox

1. Open the terminal on the left-hand side of the BitCurator desktop
2. Type "rclone config" and hit enter
3. Type "new" or "n" for new remote, i.e. new cloud service
4. Enter a name for the cloud serve (e.g. dropbox)
5. Choose the number that is listed in the terminal for cloud service you named in step 4. (e.g. Dropbox is at time of writing, 13)
6. Hit enter for questions about client id and client secret to accept the defaults
7. Type "no" and hit enter for the advanced config question
8. Type "y" and hit enter for the auto config option
9. A link will appear in the terminal if it doesn't open in a browser, highlight and copy it. Paste the link into your internet browser.
10. On the page that pops up, choose the option to authorize Rclone access
11. Return to the terminal, if you see "got code" as part of the output above, type "y" for "this is ok" and hit enter
12. Type "q" and hit enter to quit if done setting up cloud connections.

OneDrive or Sharepoint

1. Open the terminal on the left-hand side of the BitCurator desktop
2. Type "rclone config" and hit enter
3. Type "new" or "n" for new remote, i.e. new cloud service
4. Enter a name for the cloud serve (e.g. onedrive)
5. Choose the number that is listed in the terminal for cloud service you named in step 4. (e.g. OneDrive is at time of writing, 26)
6. Hit enter for questions about client id and client secret to accept the defaults
7. Choose a national cloud region for OneDrive, most likely, 1 - "Microsoft Cloud Global"
8. Type "no" and hit enter for the advanced config question
9. Type "y" and hit enter for the auto config option
10. A link will appear in the terminal if it doesn't open in a browser, highlight and copy it. Paste the link into your internet browser.
11. Return to the terminal, if you see "got code" as part of the output above.
12. Type of connection, enter a number for the type of OneDrive/Sharepoint connection. 1 for basic OneDrive Personal or Business, you will get your personal one drive account and anything shared with you. 2 for sharepoint root will give you sharepoints that are open to you. 3 you can choose enter the site url for a specific sharepoint site, this is the easiest option to find exactly what you want.
13. If you chose option 2, select the sharepoint from the list that you want to access. If you chose option 3, enter the url for the sharepoint site.
14. If the drive you selected looks good, type "y" for "this is ok" and hit enter
15. You will then get a summary of the configuration, if everything looks ok, type "y" for "this is ok" and hit enter
16. Type "q" and hit enter to quit if done setting up cloud connections.

Google Drive

1. Open the terminal on the left-hand side of the BitCurator desktop
2. Type "rclone config" and hit enter
3. Type "new" or "n" for new remote, i.e. new cloud service
4. Enter a name for the cloud serve (e.g. googledrive)
5. Choose the number that is listed in the terminal for cloud service you named in step 4. (e.g. Google Drive at time of writing is 15)
6. For application client id, the digital archivist has set this up. Contact them for access to the ID. Copy and paste that here and hit enter

7. The OAuth Client Secret, follow the same steps as 6 which will be found in the same place as the client ID.
8. For "Scope that rclone should use when requesting access from drive" choose 2 - "Read-only access to file metadata and file contents."
9. Press enter for ID of root folder
10. Press enter for Service Account Credentials JSON file path
11. Type "no" and hit enter for the advanced config question
12. "y" and hit enter for the auto config option
13. A link will appear in the terminal if it doesn't open in a browser, highlight and copy it. Paste the link into your internet browser.
14. On the page that pops up, choose the option to authorize Rclone access
15. Configure this as a shared drive - (still need to check on this)
16. Return to the terminal, if you see "got code" as part of the output above, type "y" for "this is ok" and hit enter
17. Type "q" and hit enter to quit if done setting up cloud connections.

Using Rclone

In general we use Rclone for transferring files from cloud services. When possible we also use it to confirm the fixity of the files downloaded.

Appraisal or preparing for a transfer

When transferring content from a cloud service, it may not be apparent the extent or content contained in storage. Rclone has some features which allow for basic analysis of the contents of cloud storage.

Creating a list of files and basic metadata

The `lsf` command allows for listing information about files in a machine-readable way. Particularly useful options are `-s` for size, `-t` for modified time, and `-m` for mimetype (i.e. the file type). Through this command you can create a CSV listing this information along with the paths `-p` which can be used for appraisal of the content's preservation needs, content analysis based on file names and modified dates, and to sum the size column to determine how much storage a transfer will require. Here is an example:

```
rclone lsf --csv --format ptms -R --files-only [name of remote as set up above]:[name_of_folder_or_file (if spaces in name, you can put quotation marks around this after the colon)] > path/to/csv/filename.csv
```

here is an example for Dropbox:

```
rclone lsf --csv --format ptms -R --files-only dropbox:"Radhika Nahpal INT" > /media/sf_BCShared01/processing/2022_061acc/file_list.csv
```

Finding if a transfer contains Google drive formats

When transferring content from Google Drive, there may be Google objects (Docs, Sheets, Slides, etc.) that will not be exported by Rclone in the native Drive "format" but in an equivalent format such as Microsoft Word instead. Additionally, there are some object types that cannot be exported by Rclone (such as Forms). In order to prepare for how best to export these files, it is useful to make a list of all the files in the transfer that are Google objects. This list should be included in the submission documentation of the transfer to record the original format of these files.

In order to find this information, we will use the `lsf` command similar to above but add `--drive-show-all-gdocs` to show the google docs (even those that can't be exported) and `--metadata-include` to filter for only Google object mimetypes. We will also exclude size, since Rclone cannot measure that for Google objects.

Here is an example:

```
rclone lsf --csv --format ptm -R --files-only --drive-show-all-gdocs --metadata-include "vnd.google-apps.*" googledrive:"Radhika Nahpal INT" > /media/sf_BCShared01/processing/2022_061acc/orig_in_gdrive_formats.csv
```

Copying files

Basic copying

The command to copy files is fairly simple, you specify that you want to copy the files, enter their location, and then their destination. For instance:

```
rclone copy [name of remote as set up above]:[name_of_folder_or_file (if spaces in name, you can put quotation marks around this after the colon)] [/path/to/destination/folder/originalname, i.e. processing folder, etc. If you want to retain the original folder name, enter it here, quoted if there are spaces in it]
```

Here is an example:

```
rclone copy dropbox:"Radhika Nahpal INT" "/media/sf_BCShared01/processing/2022_061acc/Radhika Nahpal INT"
```

Shared folder copying

Things shared with you in cloud services often appear in a separate section from your personal storage area. In order to access that with Rclone you will often need to add a flag for the specific service to the copy command detailed above. Here are some the flags for common services at MIT:

- Google Drive: `--drive-shared-with-me`
- Dropbox:
 - doesn't appear to always be necessary but you can use `--dropbox-shared-folders` for folders or `--dropbox-shared-files` if looking for an individual file.
 - There is no way to copy content with only an open link so you can either copy content to your personal Dropbox or ask the creator to add your email as a user of the folder.
 - For team folders, put a slash after the colon for your remote and the path can be entered after the slash. To list all the team folders available to you, do: `rclone lsd dropbox:/`
- OneDrive/Sharepoint: currently not functional but [there is a workaround](#) that could work.

Google drive copying

Google drive has some unique features that sometimes allow for or require alternative steps.

When transferring files from Google Drive (**that do not include Google objects** such as Docs, Sheets, and Slides) and additional analysis will most likely not be needed, such as a small transfer of word documents, you can direct the output of Rclone to a folder that aligns with Archivemata's standard packaging structure. This will save some work later when [preparing for Archivemata](#). Here is an example:

```
rclone copy googledrive:"Radhika Nahpal INT" "/media/sf_BCShared01/processing/2022_061acc/objects/Radhika Nahpal INT"
```

You may need to check if Google objects exist in your transfer, see this section above (link). Most common formats of Docs, Sheets, and Slides we will choose to export in open document equivalent formats. You can do this by setting the [google drive export formats](#) (the default are Microsoft Office documents). There are other options (such as PDF) described at the link above. Here is an example:

```
rclone copy googledrive:"Radhika Nahpal INT" --drive-shared-with-me "/media/sf_BCShared01/processing/2022_061acc/objects/Radhika Nahpal INT" --drive-export-formats ods,odt,odp
```

Extracting checksums

Some cloud providers have checksums stored in their system that you can extract and facilitate fixity checking. Some are unique to their system or some can be more standard types. Here is a general layout of the command to extract the checksums into a text file:

```
rclone hashsum [type of checksum] [remote source]:"folder_name or file" (same as used when copying) --output-file /path/to/output/file.txt
```

Here is an example for dropbox:

```
rclone hashsum dropbox dropbox:"Radhika Nahpal INT" --output-file /media/sf_BCShared01/processing/2022_061acc/submissionDocumentation/dropbox_checksums.txt
```

Here is an example for OneDrive or SharePoint:

```
rclone hashsum quickxor onedrive:"Radhika Nahpal INT" --output-file /media/sf_BCShared01/processing/2022_061acc/submissionDocumentation/onedrive_checksums.txt
```

Here is an example for Google Drive, because you can reuse md5 checksums in Archivemata, we can name the checksum file and store it in it's standard packaging and naming structure:

```
rclone hashsum md5 googledrive:"Radhika Nahpal INT" --output-file /media/sf_BCShared01/processing/2022_061acc/submissionDocumentation/checksum.md5
```

Google objects, such as Docs, Sheets, and Slides, do not have checksums stored in Google Drive that can be extracted. If you have any of these in the content you're transferring, they will be downloaded as regular files, but they will not have checksums in the checksum file extracted from Google Drive. In these cases, we will not reuse the checksum file we create here in Archivemata and it should be named `googledrive_checksums.txt` in a location of your choosing for later inclusion in submission documentation.

Confirming fixity

In order to confirm fixity, there are number of options:

Confirm the using the checksums you extracted in the steps above:

```
rclone checksum [checksum type] /path/to/checksum/file.txt /path/to/local_directory/of/copied_files
```

Here is an example for dropbox:

```
rclone checksum dropbox /media/sf_BCShared01/processing/2022_061acc/submissionDocumentation/dropbox_checksums.txt "/media/sf_BCShared01/processing/2022_061acc/Radhika Nahpal INT"
```

Confirm without local checksums/those that rclone generates:

```
rclone check [remote name]:[source folder] /path/to/local_copy/of/source_folder
```

Next steps and packaging files

While Rclone exports the files from the cloud provider, under most circumstances, it doesn't perform the needed packaging or analysis that will be needed for processing the files. Proceed to the [Logical Transfer](#) section for next steps in processing this content.