
RODEOS

Release 0+untagged.12.g354bba1

Manuel Holtgrewe, Dieter Beule

Jan 29, 2021

INTRODUCTION

1	Overview	3
2	Related Software	5
3	Ingest Architecture	7
4	Connecting Sequencers	9
5	Connecting Demultiplexing	11
6	Reference Implementation	13
7	iRODS Customization	15
8	Administrator Use Cases	17
9	Facility Staff Use Cases	19
10	Customer Use Cases	25

The aim of the RODEOS (Raw Omics Data accEss and Organization System) system is to facilitate the management and access to Omics raw mass data (e.g., genomics or proteomics data). The system itself is based on the [iRODS](#) ecosystem:

- [iRODS](#) for mass data storage and meta data management,
- [Metalnx](#) as a graphical user interface to [iRODS](#), and
- [Davrods](#) for WebDAV based access to the data.

This is the main entry point for the documentation. It describes the big picture and how the different components interact. Where appropriate, it references to the (external) documentation of the particular components.

OVERVIEW

This section gives an overview of the RODEOS system independent from the particular use case and supported omics data type. Figure 1.1 illustrates the overview. The central component for data and meta data storage is the RODEOS iRODS server. iRODS allows to store data in a file directory tree and also annotate folders (called *collections* in iRODS) and files (called *data objects* in iRODS) with arbitrary meta data.

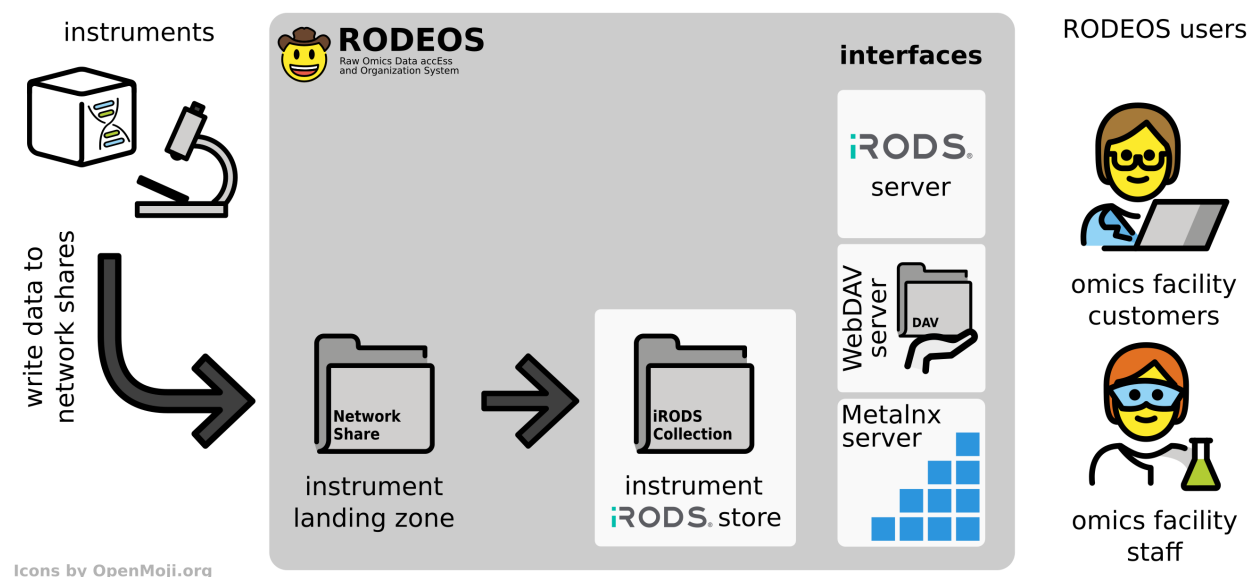


Figure 1.1: An overview of the RODEOS system. Instruments write their data to a network share (*landing zone*). The data is then imported (*ingested*) into the RODEOS iRODS server. Facility staff can use several interfaces to manage and share the data to facility customers that can use different interfaces to access the data.

Instruments and other data generation processes (such as automated data post processing steps) write the data to a particular folder for the given instrument on network file shares (so-called *landing zones*). These landing zones are monitored by automated import (so-called *ingest*) processes. The ingest processes copies the data into the appropriate collection in the iRODS server for this instrument.

RODEOS provides different interfaces to the data in the iRODS system.

iRODS Server Users can connect directly to the iRODS server with native iRODS command line tools and client libraries. This facilitates automated data download or manipulation with scripts on the command line for power users.

WebDAV Server The data is also exposed through a WebDAV server. The WebDAV protocol is supported by many graphical clients and operating systems. This allows users for graphical access to their data.

Metalnx Server Metalnx is a web-based graphical tool for accessing iRODS server functionality. This server provides omics facility staff with the required data management tools. While not directly aimed at them, facility customers can also use Metalnx functionality for accessing data that has been shared with them.

The RODEOS users are omics facility staff and omics facility customers. Omics facility staff can then use the interfaces provided by RODEOS to manage the data. This includes commonly used file and folder operations such as moving files. They can also curate meta data for data. Further, they can share data with other users for data delivery.

Omics facility customers can also use the provided interfaces to obtain the data that has been shared with them.

RELATED SOFTWARE

2.1 RODEOS Components

RODEOS Ingest Is the RODEOS component for importing raw omics data into iRODS.

2.2 Used Software

iRODS RODEOS uses iRODS for data and meta data storage.

irods_capability_automated_ingest This library is the foundation of RODEOS Ingest.

Metalnx Metalnx is a web-based user interface for iRODS. This is used for offering end users an easy-to-use front-end for managing their data in iRODS.

Davrods Davrods is a WebDAV front-end for iRODS. WebDAV is a widely used protocol for accessing data in iRODS and this can be used for graphical user interfaces and is an alternative to the iRODS iCommands (the command line user interface programs provided by iRODS).

irods-sudo-microservices. is a collection of iRODS microservices (function library for the iRODS rule language) that allow privilege escalation for normal users. This is used for allowing omics facility members to manage groups of customers.

INGEST ARCHITECTURE

Ingest is the term used for *importing data into the RODEOS iRODS Server*. RODEOS uses the iRODS *landing zone* pattern. The overall process is illustrated in [Figure 3.1](#).

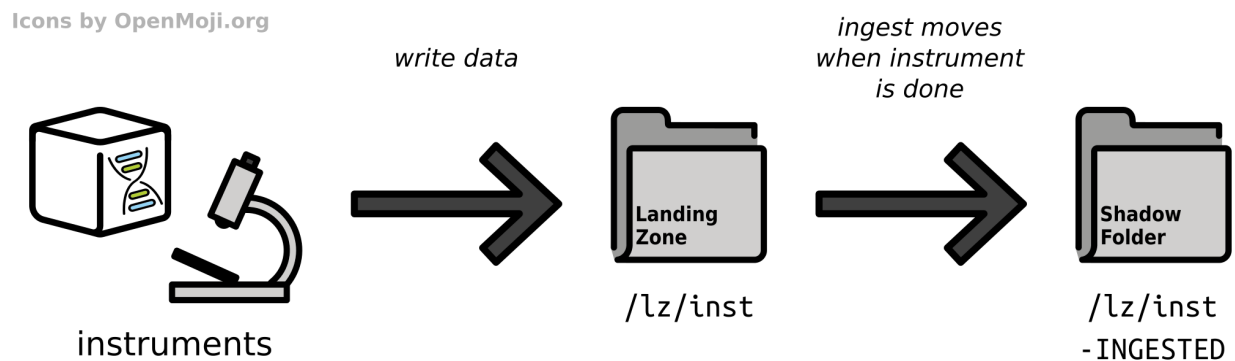


Figure 3.1: Overview of the ingest mechanism from instrument through the landing zone to the corresponding shadow folder.

That is, data is being copied or written into directories that the *ingest software component* has access to. In most cases, each landing zone folder `/${LZ}` has a corresponding *shadow folder* `/${LZ}-INGESTED`. Data items (e.g., files or directories) are written as direct members of `/${LZ}`. By the presence or content of certain files, the ingest process detects that the data item is complete. Once the data has been ingested completely and successfully into iRODS, the data item is then moved into the `/${LZ}-INGESTED` directory.

This allows using so-called machine accounts and/or enabling automatically logging into systems for instruments as common in the labs by creating a landing zone for each instrument and only sharing the `/${LZ}` folder and not the `/${LZ}-INGESTED` with the instrument driver computer. The instrument can only see the data that is currently being written and ingested. Once complete, the data is moved into a location that it cannot access.

Multi-step pipelines with multiple ingest can also be implemented. [Figure 3.2](#) gives the base call to sequence conversion for genomics facilities as an example. The sequence conversion software `bcl2fastq` reads the base call (BCL) files from a disk location that it can access (directly from the shadow directory as shown here or after staging data from the iRODS data store to the local disk). It writes into a landing zone dedicated to the generation of the sequence FASTQ files.

Another ingest process runs and transfers the data from the FASTQ landing zone into iRODS. As for all RODEOS ingest processes, the data is moved into the shadow directory for the FASTQ landing zone once complete.

Note that the ingest process is regularly running with at most one process at a given time. The ingest is implemented in a “updating” fashion similar to `rsync` (technically commonly referred to as being “idempotent”) which means that incomplete transfers are continued in the next ingest cycle until complete.

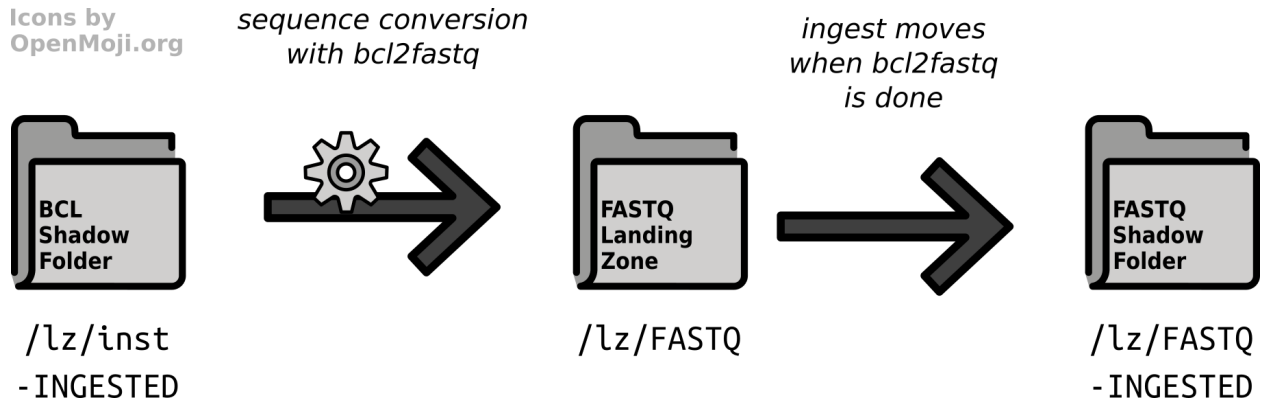


Figure 3.2: Overview of the combined process of data processing followed by ingest with sequencing base call to sequence conversion as an example.

Limitations of the ingest process include the following (the list will be updated as the authors are getting aware of more limitations).

1. In the case that an instrument does not complete (e.g., power loss or component defects), human -intervention is required to decide further actions and remove the partially written data if needed.
2. It is expected that the pause between two instrument runs (e.g., needed for a cleaning step in the lab) is sufficient to ingest data before the next instrument run. In the case that the run is started before the previous ingest step is complete, the not yet ingested data set will remain in the landing zone. The operator has to take measure to ensure that no data is overwritten.

It is common for instruments to write to a storage location where data is not moved automatically, and distinct output folder or file names are usually used for each run (e.g., incorporating the instrument ID and run number). It is thus not expected that this limitation is negligible.

The following sections give an overview and rationales for the import of the supported data types. Technical details about the ingest steps are described in the `rodeos-ingest` software package's documentation.

CONNECTING SEQUENCERS

Currently RODEOS has support for Illumina sequencing machines only. These sequencers are connected to Windows-based driver computers that write to network shares.

RODEOS uses Windows network shares to expose the landing zones to the sequencing machines. It is best practice to:

- have one account setup for each sequencing machine as these machines are usually directly accessible to any user who walks up to them,
- have one network file share for each sequencing machine that is mounted on the machines automatically.

In the case of not being able to configure file mounts through the Windows ActiveDirectory / network file share infrastructure of the organization, a Windows BAT file such as the following one can be used to mount the network share at a given drive (here U:).

Listing 4.1: File Connect RODEOS.bat

```
net use /delete u: >nul
net use u: \\<server>\<share> /user:<user>@<domain> <password>
```

Such files can be placed directly on the Windows Desktop and the instrument operator can connect the network shares before starting the sequencing. RODEOS will move the sequencing data directly after detecting that the sequencing is complete such that persons accessing the instrument can only see the currently run being written (if any). The files used for detection the completion state depending on the instrument model and device software, e.g., `RTAComplete.txt` for certain combinations.

Adding support for further device types is well possible because of the extensible nature of RODEOS. Support for mass spectrometry machines is forthcoming. An evaluation of support for PacBio or Oxford Nanopore sequencing machines will be evaluated but these devices are more tightly integrated with the vendor's processing software.

CONNECTING DEMULTIPLEXING

Demultiplexing is a data generation step similar to instruments that write data. Data is read from Illumina sequencer run directories. Ideally, the `#{LZ}-INGESTED` “shadow” landing zone directory is readable for the demultiplexing process (as in the reference implementation) but in principle the data can also be retrieved from the ingested data in RODEOS iRODS. Usually, the `bcl2fastq2` tool by the vendor Illumina is run in a custom wrapper script. The wrapper script should write a marker file indicating the completion of the demultiplexing process.

The RODEOS Ingest software comes with built-in support for our software package [Digestiflow](#) but its implementation is generic and also allows custom demultiplexing scripts with custom marker file names.

REFERENCE IMPLEMENTATION

This section describes an example/reference implementation/installation of RODEOS as deployed by [Core Unit Bioinformatics \(CUBI\)](#) at Berlin Institute of Health.

6.1 Centralized Storage System

CUBI uses a centralized storage system based on Ceph and CephFS. All data for the RODEOS system is stored below `/rodeos`.

iRODS Server The iRODS server uses the server `/rodeos/irods-vault` for storing its data. It only has this path mounted such that it cannot access anything else from the central file system.

HPC System The HPC system has the path `/rodeos/lz` mounted and unix groups and permissions are used to provide the data generation units with appropriate access to their data. The iRODS data vault is only available to the iRODS server.

6.2 Landing Zones

For instruments such as sequencing devices, the landing zones are implemented as follows.

Landing Zones For each data generation unit `UNIT` and each instrument (or data generation process) `INST`, a folder `/rodeos/lz/UNIT/INST` is created. These folders are exported via Windows SMB/CIFS network shares and mounted on the instrument driver computers. The instruments write into these directories.

Shadow directories Further, the folder `/rodeos/lz/UNIT/INST-INGESTED` exists for each group and instrument and serves as the shadow folder. This folder is not exposed to the instrument but made available to the data generation unit with appropriate unix permissions.

For data processing that runs on the HPC system, the data generation (through processing) is implemented as follows. Both landing zones and shadow directories are also managed below the `/rodeos/lz` directory as above but they are not shared via the network. Instead, there are dedicated Unix users for the data processing (that is separate from the instrument's Unix user) and Unix groups and permissions are used such that the processing user can write into the landing zone but cannot access the shadow folder.

6.3 Data Ingest

Data ingest is implemented using one dedicated ingest server that has `/rodeos/lz` mounted. There is one dedicated Unix account for the ingestion of each data generation source, that has corresponding a iRODS account and is permanently authenticated as this account. The ingest process runs as this Unix user and monitors the corresponding landing zone, ingests the data into iRODS, and moves the data from the landing zone into the shadow folder when done.

6.4 Digestiflow

Demultiplexing of genomics data is used using the Digestiflow system developed at CUBI. Genomics facility staff uses the Digestiflow server as documented in the [Digestiflow documentation](#).

Demultiplexing and ingest of data is done using a periodically running job on the HPC system. Both jobs run as the same Unix user, one for each genomics facility. There is one `FASTQ` landing zone (with corresponding shadow directory) for each facility and data is ingested as in the general case.

6.5 Data Delivery

Data delivery has been implemented as described in the use case section. For delivery, it is assume that the genomics unit staff creates a dedicated collection for each project (the definition of a “project” depends on the genomics unit, e.g. a service order). Data from one or more sequencing runs (either a whole or parts of a run) can be delivered by moving or copying it into that collection in RODEOS iRODS.

The delivery process is summarized as below. Once the data is ready:

- **genomics facility staff**
 - moves the data into the project-based delivery folder
 - ensures that the destination user has the appropriate permissions to read the data
 - sends out an email with the information on how to access the data to the customer
- **customer**
 - receives the email with information
 - if necessary reads the documentation provided on a public server about the different access options
 - downloads the data using the provided iRODS commands or via WebDAV

IRODS CUSTOMIZATION

This section describes the iRODS customizations that have been installed / implemented.

7.1 Rule Adoptions

- By default, no home directory is created for newly created users. This is a deviation from the standard iRODS behaviour.

7.2 Microservices

The software `irods-sudo-microservices` has been installed on the iRODS provider (catalogue) server. This allows to implement rules with privilege escalation.

7.3 AVU Prefixes

Generally, the RODOES system uses the prefix `rodeos::` for AVU (attribute value unit) triples.

The RODEOS Ingest subsystem uses the prefix `rodeos::ingest` for meta data annotation and state management.

To allow facility members to manage groups of customers, AVU triples with the attribute name `rodeos::sudo::group-prefix` are created. Such users can then invoke the `msiSudoUserAdd()` microservice to manipulate groups whose name starts with the given prefix. Of course, good care has to be taken for such prefixes to be unique, generally group names are `#{UNIT}::#{CUSTOMER}` with short identifiers of the unit and the customer group.

7.4 Rules

The following rules have been implemented for the `irods-sudo-microservices` package.

acPreSudoGroupAdd Allow users to add groups with the name that starts with the value of any value of the `rodeos::sudo::group-prefix` attribute.

acPreSudoGroupRemove Allow users to remove groups with the name that starts with the value of any value of the `rodeos::sudo::group-prefix` attribute.

acPreSudoGroupMemberAdd Allow users to add members to groups with the name that starts with the value of any value of the `rodeos::sudo::group-prefix` attribute.

acPreSudoGroupMemberRemove Allow users to remove members from groups with the name that starts with the value of any value of the `rodeos::sudo::group-prefix` attribute.

7.5 Custom Scripts

7.5.1 RODEOS Facility Helper Scripts

RODEOS ships with a number of Bash scripts that help facility staff in the management of users:

rodeos-cli-group-list Lists existing groups that the current user can manager.

rodeos-cli-group-create **GROUP_NAME** Create a new group with the given name.

rodeos-cli-group-remove **GROUP_NAME** Delete group with the given name.

rodeos-cli-group-member-add **GROUP_NAME** **USER_NAME** Add a user with the given name to the given group.

rodeos-cli-group-member-remove **GROUP_NAME** **USER_NAME** Remove a member from a group.

ADMINISTRATOR USE CASES

This section describes the use cases for the administrator.

8.1 Create a new Unit

In this use case the RODEOS (iRODS) administrator creates a new data generation unit. The steps are as follows:

1. Create a new group in the underlying unix user/group management system and add all users to it.
2. Create a user for the sequencers in the unit's home organization user directory and make them known in the HPC system as well.
3. Setup the landing zones on the storage system and assign appropriate Unix ownership and permissions.
4. Setup the ingest users in the unit's home organization user directory.
5. Setup the ingest server with ingest jobs.

Steps 1-5 can be automated with the RODEOS installation Ansible playbooks.

6. Create a group for the unit members in iRODS and add them to the group.
7. For the group members that should be able to manage groups for the customers, adjust the meta data attribute `rodeos::sudo::group-prefix` and inform users about the prefix to use.

FACILITY STAFF USE CASES

This section describes the use cases for the facility staff. As an example, the case of a sequencing facility is used where data generation equals performing a sequencing run. This step can be adjusted appropriately for other data generation unit types.

9.1 Group Management for Customers

In this use case, facility staff members want to manage their customers in groups in iRODS. For new customers they want to create a new group or for retired customers they want to remove groups. At any other time they want to add/remove users to/from groups.

9.1.1 Prerequisites

- An account must have been properly setup for the user by the RODEOS / iRODS administrator. This includes allowing access to privilege for the administration of groups.
- The user must know the prefix for the groups that they can manage (e.g., `gen-cha::cust::`).
- The user must have setup iRODS iCommands correctly and have configured `~/.irods/irods_environment.json` properly.
- The user must have the RODEOS facility staff helper scripts installed.

9.1.2 Steps

- Authenticate with the iRODS server using `iinit`.
- Use the *RODEOS Facility Helper Scripts* to manage the groups.

9.2 Create New Project Collection

In this use case, a facility staff member creates a new project collection (“collection” is the iRODS term for a folder in the iRODS storage system). Such a project collection serves as the location for resulting data to share with the customer. This collection will have read permissions set for the customer group/user recursively and permission inheritance is enabled. This way, customer users can download the data once it has been provided.

9.2.1 Prerequisites

- A group has been setup for the customers if access needs to be given based on more than one user.

9.2.2 Steps

Use Metalnx to

- create a new folder in the `projects` collection of the facility
- make sure that `inheritance` is enabled for the collection and use `Apply recursively` to apply this to all existing sub folders
- configure permission and add a new `ACL` for the customer group or user with the `READ` permission, make sure to select `Apply to subcollections and files` such that existing and data placed in the directory afterwards gets the correct permissions setup

9.3 Perform Sequencing Run

In this use case, facility staff members start a sequencing run into the landing zone provided by RODEOS.

9.3.1 Prerequisites

- The user for the sequencer and ingest process must exist.
- Ingest must have been setup appropriately.

9.3.2 Steps

- Connect network drive to the network share for the sequencer if necessary.
- Write data to this network share.
- Wait until data generation is complete.
- The output folder will be moved into the shadow landing zone folder afterwards.

9.4 Perform Sequence Conversion

In this use case, facility staff members start the conversion process from base calls to sequences (`bc12fastq`) that is also sometimes referred to as “demultiplexing”.

9.4.1 Prerequisites

- Digestiflow must have been setup correctly for the sequencer for which demultiplexing should be performed.
- Sequencing should have finished.

9.4.2 Steps

- Start demultiplexing as documented in the Digestiflow documentation.
- Wait for demultiplexing to finish.
- The resulting data will appear in the `FASTQ` collection in iRODS.
- The meta data `rodeos::ingest::status` will be set to `complete` once done.

9.5 Deliver Conversion Results

In this use case, facility staff wants to provide sequencing results to customers. These could be sequences in FASTQ format and/or archives from raw BCL data such as tarball files created by Digestiflow.

9.5.1 Prerequisites

- Ideally, a project collection has been created for file delivery to the customer.
- Permissions have been created appropriately as described in *Create New Project Collection*.

9.5.2 Steps

Use Metalnx to:

- create an output collection in the project collection, e.g., named like the flowcell
- go to the folder with the digestiflow demux results
- mark the files and/or folders to move
- move them into the output directory
- **in the case that additional data is required for delivery (e.g., manually created QC reports)**
 - the facility staff generates the reports, and
 - copies them into the project folder
- notify the customer about the arrival of new data and instructions how to access the data

9.6 Provide Raw Data Access

In this use case, facility staff wants to provide direct access to raw data.

9.6.1 Prerequisites

- None.

9.6.2 Caveats

- It is best practice to have only one location from which data is shared.
- Raw data should probably not be shared even read-only.
- For BCL raw data, providing archives as created by Digestiflow are more efficiently shared than the tens of thousands of files in a run folder.

9.6.3 Steps

- Use Metalnx to set the appropriate permissions on the raw data folder.
- Share the path to this folder with the customer together with instructions on how to access the data.

9.7 External Customer Delivery

In this use case, facility staff wants to deliver data to external customers.

9.7.1 Prerequisites

- The customer must be provided with an identity in the host organization's user account directory (e.g. ActiveDirectory). The account can be limited but at least a user name and password must exist. The rationale is that for the transfer of human data which will be necessary in the general case, it will be required that the receiving party is a natural human being whose identity is verified, e.g., by the human resource department.

9.7.2 Steps & Caveats

Once the customer has an identity in the host organization's user account directory, the delivery process is very similar to the use cases *Deliver Conversion Results* and *Provide Raw Data Access* apply. However, facility staff will have to mark the result iRODS collection (using iRODS meta data through the graphical Metalnx) to be delivered through a particular server that is also reachable from the outside.

9.7.3 Status

This use case has been registered but not implemented yet. It is expected to be implemented at a later milestone. Until then, other delivery means have to be used with preexisting means.

CUSTOMER USE CASES

This section describes use cases for facility customers.

10.1 iRODS Data Access

In this use case, a customer downloads data that has been delivered to them by the facility staff (cf. *Facility Staff Use Cases*) via the **iRODS iCommands**.

10.1.1 Prerequisites

- The user must have the iRODS iCommands command line tools installed. These are only available for Linux and MacOS. Windows users should use the WebDAV protocol.
- The user has knowledge of the Linux/MacOS command line. Inexperienced users are recommended to use the WebDAV protocol with a graphical client.
- The user must be able to connect to the iRODS server. For the server operated by CUBI, the client must be in the Charite/MDC/BIH networks or have appropriate VPN access.

10.1.2 Steps

- Perform the iRODS setup, in particular creating a proper `~/.irods/irods_environment.json` file.
- Successfully authenticate to the iRODS server with `iinit`.
- Use `irsync -rkv i:${SOURCE} ${DEST}` to download the data from the `${SOURCE}` collection in iRODS as obtained from the data generation facility to the local destination `${DEST}`.

10.2 Graphical WebDAV Data Access

In this use case, a customer downloads data that has been delivered to them by the facility staff (cf. *Facility Staff Use Cases*) via the **WebDAV protocol using a graphical client**.

10.2.1 Prerequisites

- The user must have a graphical client installed that can use the WebDAV protocol. Recommended free software is:
 - WinSCP for Windows
 - Cyberduck for Mac Os X
 - The usual file browsers for Linux.
- The user must be able to connect to the iRODS server. For the server operated by CUBI, the client must be in the Charite/MDC/BIH networks or have appropriate VPN access.

10.2.2 Steps

- Connect to the RODEOS WebDAV server and login to the system.
- Go to the location `${SOURCE}` where the data for the user resides.
- Download the data through the graphical client's functionality (probably drag and drop).

10.3 LFTP WebDAV Data Access

In this use case, a customer downloads data that has been delivered to them by the facility staff (cf. *Facility Staff Use Cases*) via the **WebDAV protocol using the command line client lftp**.

10.3.1 Prerequisites

- The user has knowledge of the Linux/Mac command line. Inexperienced users are recommended to use the WebDAV protocol with a graphical client.
- The user must have lftp installed. The software is only available on Linux/MacOS so such an operating system is a prerequisite for installing lftp itself. Please install lftp with your Linux package manager or a tool like Homebrew for MacOS for installing lftp. If this poses a problem we recommend using one of the graphical WebDAV clients as described above.
- The user must be able to connect to the iRODS server. For the server operated by CUBI, the client must be in the Charite/MDC/BIH networks or have appropriate VPN access.

10.3.2 Steps

- Connect to the WebDAV server using the user's account.
- Download the data from the `${SOURCE}` collection where the data for the user resides using `mirror ${SOURCE} ${DEST}` to the local destination directory `${DEST}`.