

# Using iCommands

## IMPORTANT NOTE ABOUT UPGRADING TO 4.1

iCommands 4.1.10, the latest supported version for CyVerse services, has been released by iRODS. For a list of testing done on affected services, see [Upgrading to iRODS 4.1](#).

iCommands is a collection of commands for Linux and Mac OS operating systems that are used in the iRODS system to interact with the CyVerse Data Store. Many commands are very similar to Unix utilities. For example, to list files and directories, in Linux you use `ls`, but in iCommands you use `ils`.

While iCommands are great for all transfers and for automating tasks via scripts, they are the **best choice for large files** (> 2 GB each) and for **bulk file transfers** (many small files). For a comparison of the different methods of uploading and downloading data items, see [Downloading and Uploading Data](#).

iCommands can be used by CyVerse account users to download files that have been shared by other users and to upload files to the Data Store, as well as [add metadata](#), change permissions, and more. Commonly used iCommands are listed below. Follow the instructions on [Setting Up iCommands](#) for how to download and configure iCommands for your operating system.

**A CyVerse account is not required to download a public data file via iCommands.** To see instructions just for public data download with iCommands, see the [iCommands section on Downloading Data Files Without a User Account](#).

Before you begin, you may want to watch a CyVerse [video about iCommands](#).

## About setup and configuration

Regardless of whether you want to download and upload data as a CyVerse user, or download public data without an account as an anonymous user, iCommands must be set up and configured on each user's local machine or server. **For information on downloading and setting up iCommands for your operating system, see [Setting Up iCommands](#).**

**For information on downloading public data files anonymously, see [Downloading Data Files Without a User Account](#).** To share data with a user, the user at minimum must have Read permission to the parent directory that contains the file and any other parent folders within the folder hierarchy.

In addition to using iCommands to share data from the command line, CyVerse users can upload and download data within the Discovery Environment. For more information, see [Sharing Data Files and Folders Via the Discovery Environment](#).

## Commonly used iCommands

These sample commands are entered from the command line once you have installed and configured iCommands on your computer.



### This is not a comprehensive guide for how to use iCommands

This section offers the most commonly used commands used with CyVerse data. If you are not comfortable using the command-line interface, ask your IT professional to assist you.

For more information and a complete list of all iCommands, enter the iCommand and `-h` in the terminal window to view the related Help information. More information is available by searching on "icommands" at <http://irods.org/documentation/>.

For more information on using these and other iRODS commands, see the [iRODS Documentation](#) on the iRODS website, or enter the iCommand and `-h` to view the help file in a terminal window.

Enter any of the following iCommands in a command line.

Command	Description (For help, enter the command followed by -h)
icd	<p>Changes the working directory.</p> <p>▼ Example: <code>icd</code></p> <pre>bash-3.2\$ icd tmp bash-3.2\$ ils /iplant/home/yourusername/tmp:  2011_UA_PLS340_Genomics.key  2011_UA_PLS340_Genomics.key.pdf  AllHandsAug162011.key  gff_annotation_loader.pl</pre>
ichmod	Gives access to another user.
ichmod read	Grants read-only permission level for specified user to selected file or folder.
ichmod write	Grants read and write permission level for specified user to selected file or folder.
ichmod own	Grants full ownership permission level for specified user to selected file or folder.
ichmod null	Removes the permission level for the user to the file or folder.
iexit full	<p>Logs off/disconnects from the Data Store.</p> <ul style="list-style-type: none"> <li>To log out of the Data Store (<b>especially important if on a shared or public computer</b>), enter <code>iexit full</code></li> <li>To log in again on the same device, enter <code>iinit</code></li> </ul> <p>After you are successfully connected, you do not need to reenter your password to use the CyVerse Data Store from that computer unless you log out.</p>
iget	<p>Downloads the file/directory from iRODS to local device.</p> <p>Example:</p> <pre>bash-3.2\$ iget -K file_name.txt</pre> <p>In the above example, the <code>-K</code> flag is included to ensure data isn't corrupted during data transfer via iCommands, which can happen anytime data are transferred across the internet. CyVerse highly recommends that you include the <code>-K</code> option with both <code>iget</code> and <code>iput</code>. This causes iCommands to verify that transferred files weren't corrupted during the transfer by comparing the Checksums computed before and after the transfer.</p>

iinit	<p>Initializes and starts the connection to iRODS.  <a href="#">v Example: iinit</a></p> <pre>bash-3.2\$ iinit One or more fields in your iRODS environment file (.irodsEnv) are missing; please enter them. Enter the host name (DNS) of the server to connect to:data.iplantcollaborative.org Enter the port number:1247 Enter your irods user name:yourusername Enter your irods zone:iplant Those values will be added to your environment file (for use by other i-commands) if the login succeeds. #* Enter your current iRODS password: bash-3.2\$=</pre>
iinit -h	Reports the client software version.
imiscsvrinfo	Reports the server version.
ils	<p>Lists the contents of the current working directory.  <a href="#">v Example: ils</a></p> <pre>bash-3.2\$ ils /iplant/home/yourusername: .bashrc ._.DS_Store .DS_Store .i-commands-auto.bash C- /iplant/home/yourusername/analyses C- /iplant/home/yourusername/backup C- /iplant/home/yourusername/collections C- /iplant/home/yourusername/data C- /iplant/home/yourusername/tmp C- /iplant/home/yourusername/utilities</pre>
ils -A	<p>Lists the directory permissions.  <a href="#">v Example: Directory permissions</a></p> <pre>? ils -A /iplant/home/user/directory /iplant/home/user/directory: ACL - g:rodsadmin#iplant:own user#iplant:own de-irods#iplant:own Inheritance - Disabled loupot-trimmed.fasta ACL - rodsadmin#iplant:own user#iplant:read object 2oupot-trimmed.fasta ACL - rodsadmin#iplant:own user#iplant:write object C- /iplant/home/user/directory/subdirectory</pre>

mkdir	<p>Creates a new directory.</p> <p>▼ Example: mkdir</p> <pre>bash-3.2\$ mkdir NewDataDirectory</pre>
iput	<p>Uploads the file/directory from the local device to iRODS.</p> <p>Example:</p> <pre>bash-3.2\$ iput -K file_name.txt</pre> <p>In the above example, the <code>-K</code> flag is included to ensure data isn't corrupted during data transfer via iCommands, which can happen any time data are transferred across the internet. CyVerse highly recommends that you include the <code>-K</code> flag with both <code>iget</code> and <code>iput</code>. This causes iCommands to verify that transferred files weren't corrupted during the transfer by comparing the Checksums computed before and after the transfer.</p> <p>See more iput examples <a href="#">below</a>.</p>
ipwd	<p>Shows the name and path of the current remote folder.</p> <p>▼ Example: ipwd</p> <pre>bash-3.2\$ ipwd /iplant/home/yourusername/tmp</pre>
irm	<p>Moves a file to the trash.</p> <p>▼ Example: irm</p> <pre>bash-3.2\$ irm file_name.txt</pre>
irm -f	<p>Deletes a file.</p> <p>▼ Example: irm -f</p> <pre>bash-3.2\$ irm -f file_name.txt</pre>
irm -r	<p>Moves a folder to the trash.</p> <p>▼ Example: irm -r</p> <pre>bash-3.2\$ irm -r file_name.txt</pre>
irm -fr	<p>Deletes a folder.</p> <p>▼ Example: irm -fr</p> <pre>bash-3.2\$ irm -fr file_name.txt</pre>

irsync	<p>Synchronize a <b>folder from a local directory</b> (e.g., your computer) to a directory in iRODs (or the CyVerse Data Store). The <code>-l</code> flag can be used to investigate if the directory needs to be synchronized. The <code>-s</code> flag can be used to use file size rather than checksums to determine if the directories are synchronized.</p> <pre style="border: 1px solid black; padding: 10px;">bash-3.2\$ irync -r local_directory_name i:data_store_directory_name</pre>
--------	---

## Reliability and troubleshooting tips and tricks

If you are having problems with reliability, try the following command options:

Purpose	Command
Allows <b>bulk</b> uploads for transferring many small files at once. <b>Cannot be used with <code>--lfrestart</code>.</b>	<code>-b</code>
Includes a folder or directory.	<code>-r</code>
Renews the socket connection every 10 minutes.	<code>-T</code>
Lists the number of retries attempted in case of failure/error.	<code>--retries</code>
Saves the progress to a file so that only the files that were not sent successfully are sent when you restart a failed upload. Any filename may be used for the checkpoint-file.	<code>-X &lt;checkpoint-file&gt;</code>
Saves the progress to an <b>individual</b> file so only the unsent portion of the file is sent when you restart the upload. <b>Cannot be used when using <code>-b</code> for a bulk upload.</b> Any filename may be used for the <code>checkpoint-lf-file</code> .	<code>--lfrestart</code> <code>&lt;checkpoint-lf-file&gt;</code>
Provides progress feedback.	<code>-P</code>
Provides more information about a command.	<code>-V</code>
The versions must be the same on your client and on the server. <ul style="list-style-type: none"> <li>Finds the version of iCommands on your system.</li> <li>Finds the version of iCommands on your server.</li> </ul>	<ul style="list-style-type: none"> <li><code>iinit -h</code></li> <li><code>imiscsvrinfo</code></li> </ul>

## Troubleshooting a failed DE analysis that uses a file uploaded via iCommands

If the file used in the failed analysis was uploaded to the Data Store via iCommands, verify that the filename does not contain any special characters or spaces. This has been known to cause analyses to fail. For more information, see [Using Special Characters in the DE](#) and [Renaming a Data File or Folder in the DE](#).

## Examples

### Bulk files transfer example

```
iput -K -P -b -r -T --retries 3 -X checkpoint-file
my-files-to-transfer/
```

## Single large file transfer example

```
iput -K -P -T --lfrstart checkpoint-lf-file
my-file-to-transfer.txt
```

In the above examples:

- **-K**: To ensure data isn't corrupted during data transfer via iCommands, which can happen anytime data are transferred across the internet, CyVerse highly recommends that you include the **-K** option with `iget` and `iput`. This causes iCommands to verify that transferred files weren't corrupted during the transfer by comparing the Checksums computed before and after the transfer.
- **my-files-to-transfer/** is the example name of the directory or folder for bulk transfers.
- **my files-to-transfer.txt** is the example name for single file transfers.
- Any filename may be used for the **checkpoint-file**.

### Helpful Links

#### On This Page:

- [About setup and configuration](#)
- [Commonly used iCommands](#)
- [Reliability and troubleshooting tips and tricks](#)
- [Troubleshooting a failed DE analysis that uses a file uploaded via iCommands](#)
  - [Examples](#)

#### Related Pages:

- [Setting Up iCommands](#)
- [Sharing Data from the Command Line \(iCommands\)](#)
- [Adding Metadata to a File Using iRODS imeta \(Metadata\) Commands](#)
- [Downloading Data Files Without a User Account](#)
- [Upgrading to iRODS 4.1](#)