

How to use: RAIDFS

by Team 07

1 RendezVous

Import the source code into your favorite IDE. Export it into a Runnable Jar with the main class being `MainRendezVous.java`.

Then run the jar like `java -jar rendezvous.jar`

Press **Enter** to shut down the RendezVous.

Note: You cannot run it with `'& '` at the end to run it in background.

2 Peer

Export the project as a runnable jar with the main class being `Mishell.java`.

Run the jar the same way as the previous `.jar`, only this time you have the following (optional) argument :

- `seed_address` specifies the address of the main seed, default being `icdatasrv2.epfl.ch`

More commands will be coming when the DFS will support secure connections.

The possible commands are :

1. **connect** : Connects to the RendezVous seed in order to enter the DFS network. When your peer connects, it tries to fetch the neighbours, and ask them for their current index of the DFS. If it succeeds, it can perform the following operations:
2. **ls folder_path** : lists the file in the supposed DFS folder. Only lists it if it's in our `meta.xml` file.
3. **put file_path dfs_folder_path** puts the specified local file (path starting from the folder where the `.jar` is launched) on the DFS, into the specified folder (absolute path on the DFS).

Note: to put on the DFS root, specify `'/'` as a folder.

4. **rm dfs_file_path** removes the specified file from the DFS.

5. **quit** shuts the peer and disconnects it properly from the JXTA network.

You will not be able to perform any operation on the DFS until you load the current index of the DFS, fetched from one of your peers.

In the case of bootstrapping, you have to connect at least two peers to the DFS, they will fetch the empty DFS index from each other, and then you can start putting/removing files

When you run the jar, several resources get created in the folder you launch it from :

- **.raidfs_data/** hidden folder used to store the chunks for each file on the DFS.
- ***name*/** a folder with the name your peer owns on the JXTA network. It is a 4-bytes integer randomly generated on start. It is used to store the JXTA related resources. Nothing interesting is in there.
- **meta.xml** XML file containing the index of the DFS. It is updated in real time while the Peer is running.