

```

import java.io.*;

public class CurlClient implements FilenameFilter {

    String searchPattern = "";

    public static void main(String args[]){
        //Syntax Check, returns usage of not correct
        if (args.length != 5){
            System.out.println("Usage: java CurlClient <homespace>
<directory containing files> <filepattern to upload> <fileType
GNSS/GNSSV2/SLR/DORIS/VLBI/MISC> <fileContentType>");
            System.out.println("For file content type the following
values are valid");
            System.out.println("\t'Data' , 'Products' , 'Misc'");
            System.out.println("");
            System.out.println("\tNo values are needed for VLBI");
            System.exit(0);
        }

        //Creates new instance of CurlClient with all the command line
arguments being passed
        // args[0] = homespace (where cookie file should be located
        // args[1] = directory containing data files to be uploaded
        // args[2] = pattern all the file names contain. Used to not
upload extra files
        // args[3] = fileType (values can be
GNSS/GNSSV2/DORIS/SLR/VLBI/MISC)
        // args[4] = data parameters (depends on type of data, see
documentation)
        new CurlClient().runClient(args[0],args[1],
args[2],args[3],args[4]);
    }

    public void runClient(String cookieLocation, String fileDirectory,
String searchPattern, String fileType, String fileContentType){
        try {
            this.searchPattern = searchPattern;

            //Defines the loginCmd, included location of
cookiesFile
            //May need to include full path to cURL if not found

            String loginCmd = "curl -n -L -b " + cookieLocation +
File.separator + ".urs_cookies -c " + cookieLocation + File.separator +
".urs_cookies http://cddis-basin.gsfc.nasa.gov/CDDIS_FileUpload/login";

            //Set the directory containing the upload files to the folder
variable
            File folder = new File(fileDirectory);
            //Filter files in directory, store only those that match pattern
in array
            String[] allFilesInThatFolder = folder.list(this);

            //Build the fileUpload options
            //Creates a StringBuffer object that will contain the file type
specific options

```

```

        StringBuffer uploadOptions = new StringBuffer();

        uploadOptions.append(" -F \"fileType=\"" + fileType + "\"");
        uploadOptions.append(" -F \"fileContentType=\"" + fileContentType +
"\");

        //Begin building the upload command
        StringBuffer uploadCmd = new StringBuffer();

        //Begin building upload command
        uploadCmd.append("curl -X POST -b " + cookieLocation +
File.separator + ".urs_cookies -c " + cookieLocation + File.separator +
".urs_cookies -n ").append(uploadOptions);

        //Include all files that match in upload command
        for (int i=0;i<allFilesInThatFolder.length;i++){
            uploadCmd.append(" -F \"file[]=@" + fileDirectory +
allFilesInThatFolder[i] + "\"");
        }

        //Append the URL for the upload web service
        uploadCmd.append(" http://cddis-
basin.gsfc.nasa.gov/CDDIS_FileUpload/upload/");

        //Creates file named cmd.txt in fileDirectory containing the
commands to login and upload files
        //This is needed to deal with some quirks in how the Java exec
command works and the special characters needed by the cURL command

        PrintWriter writer = new PrintWriter(fileDirectory +
File.separator + "cmd.txt", "UTF-8");
        writer.println(loginCmd);
        writer.println(uploadCmd.toString());
        writer.close();

        //Executes the file and outputs the results to standard out
        Process p = Runtime.getRuntime().exec(fileDirectory +
File.separator + "cmd.txt");
        BufferedReader in = new BufferedReader(new
InputStreamReader(p.getInputStream()));
        String line = null;
        while ((line = in.readLine()) != null) {
//            //The output could be handled differently, this is the
simplest
            System.out.println(line);
        }

    } catch (Exception e) {
        e.printStackTrace();
    }
}

//Method required to filter by file names
@Override
public boolean accept(File directory, String fileName) {

```

```
        if (fileName.contains(this.searchPattern))
return true;
    }
    return false;
}
}
```