

# Pcommands Installation and Usage Handout

## 1. Installation

- **Download the package**

Let's first get the source of Pcommands tool from PetaShare website:

go to:

<http://www.petashare.org>

click 'Downloads' from the menu on the right side.

click to 'pcommands package v.2.0'

OR,

you can click the link below, or copy and paste it to address bar of your browser to start to download the package...

<http://www.cct.lsu.edu/~akturk/petashare/downloads/pcommands/pcommands-2.0.tar.gz>

save the package in a convenient place.

- **Decompress the package**

go to the directory where you downloaded Pcommands package.

```
$ cd <parent directory of Pcommands>
```

then, run the following command to decompress it...

```
$ tar xvfz pcommands-2.0.tar.gz
```

this should show some output and create a directory called pcommands-2.0

next, go into the pcommands-2.0 directory:

```
$ cd pcommands-2.0
```

- **Run setup script to compile and install Pcommands**

```
$ ./setup
```

this should take a while to finish...

after this , you need to source new environment variables to access pcommands conveniently.

```
$ source ~/.pcommands/.bashrc.pcommands
```

you should source the appropriate file according to your shell:

if you are using bash: ~/.pcommands/.bashrc.pcommands

if you are using C shell: ~/.pcommands/.cshrc.pcommands

if you are using TC shell: ~/.pcommands/.tcshrc.pcommands

if you are using Korn shell: ~/.pcommands/.profile.pcommands

OR,

you can open a new terminal to start play with Pcommands directly...

Note that, binary files of pcommands are located in:

```
<pcommands home directory>/bin
```

## 2. Authentication

- **Setting Up Account and Password**

Pcommands package comes with default account called 'testuser'.

If you do not have existing PetaShare account, you can use 'testuser' account for testing purposes. However, this account is not safe place to put your essential data since it is accessible by anyone.

You can set up PetaShare account by running 'pchangeuser' command:

```
$ pchangeuser
```

It will ask you PetaShare username. For this tutorial you should type the given username and password.

```
> Enter Your User Name:
```

```
userXX
```

```
> Enter your current Petashare password:
```

```
passwordXX
```

After providing username and password, you are ready to access your allocation.

You need to use 'pchangeuser' command only once, it will keep your authentication information in scrambled format for further accesses.

**Note:** It is first command that you need to run when you get a real PetaShare account to activate it and set up your new credentials.

### 3. Usage Example

- **Listing Directory**

After setting your account, you may want to access your allocation. Your allocation is represented in PetaShare namespace as:

```
/tempZone/home/userXX
```

You should run the following command to list your home directory in PetaShare:

```
$ pls
```

It will show you the content of your home directory which is empty at this time.

Similarly, you can look at your current working directory in PetaShare by running the following command:

```
$ ppwd
```

that, will show *'/tempZone/home/userXX'*

- **Creating a directory**

This is one of the most basic operation that you may want to do. You can create directories by running the following command:

```
$ mkdir directoryName
```

this will create a directory called *'directoryName'* in your current PetaShare directory. You may want to use absolute path like this:

```
$ mkdir /tempZone/home/userXX/directoryName
```

- **Uploading a file into PetaShare**

Here, we will try to upload a single file, and a directory that contains multiple files(requires recursive copy, similar to 'cp'):

You can use the script to create dummy files for this demo. Script is available at:

<http://www.cct.lsu.edu/~akturk/petashare/tutorial/createDummyFile>

Alternatively, you can use your existing files in your system.

To upload a single file into PetaShare run the following command:

```
$ pput fool.dat directoryName/
```

or,

```
$ pput foo1.dat /tempZone/home/userXX/directoryName/
```

This will copy your *foo1.dat* into */tempZone/home/userXX/directoryName*. You can verify *foo1.dat* has been copied into */tempZone/home/userXX/directoryName/* by running:

```
$ pls /tempZone/home/userXX/directoryName/
```

**Note:** Upto that point, we do not talk about where the *foo1.dat* will physically be located. By default, all operations are taking place in default PetaShare site. This site is 'lsu' by default. If you want to change the default PetaShare site, you just need to run the above command in the following format:

```
$ pput foo1.dat /petashare/tulane/tempZone/home/userXX/directoryName/
```

this commands explicitly tells that, Pcommands should put file *foo1.dat* into the PetaShare resource in TULANE.

After specifying the site that you want to connect by using the following syntax:

```
/petashare/SITE_TO_BE_CONNECTED/tempZone/home/userXX
```

*SITE\_TO\_BE\_CONNECTED* becomes your default PetaShare site to be connected, so you do not need to specify it explicitly again.

Now, we want to upload a directory to PetaShare that contains many files:

```
$ pput -r FOO_DIR directoryName/
```

or,

```
$ pput -r FOO_DIR /tempZone/home/userXX/directoryName/
```

that will create *FOO\_DIR* in '*directoryName*' and copy all the files in *FOO\_DIR*.

- **Useful Options for pput**

If you are uploading multiple files into the PetaShare, you may want to use 'restart' file in case of failure during the uploading process. This file keeps the record of last successfully uploaded file, so you can restart uploading process from that file, not from scratch.

To use restart file we need to use -X option:

```
$ pput -X restartFileName -r FOO_DIR  
/tempZone/home/userXX/directoryName
```

If any failure occurs during the uploading process, run the above command again to restart the process from last successfully sent file.

Another option is `-f` option that allows you to overwrite the existing files, without getting any error message.

- **Downloading a file from PetaShare**

It is very similar to uploading a file to PetaShare. Let's say you want to download a file called *foo1.dat* from `/tempZone/home/userXX/directoryName` to home directory of your computer. We will use `pget` command to do so.

```
$ pget directoryName/foo1.dat ~/
```

or,

```
$ pget /tempZone/home/userXX/directoryName/foo1.dat ~/
```

Similarly, if you want to download a directory from PetaShare, you need to use `-r` option.

```
$ pget -r directoryName ~/
```

or,

```
$ pget -r /tempZone/home/userXX/directoryName ~/
```

- **Other commands**

There are many other commands in Pcommands package that allow users to manage their data. Other useful commands are:

**prm:** remove file or directory from PetaShare

**prsync:** to synchronize the copies of files

**prepl:** to replicate a file

**pchmod:** to change the access permissions of file

etc.

We skip these commands here to save time; however, please feel free to ask any question regarding them to [petashare-users@cct.lsu.edu](mailto:petashare-users@cct.lsu.edu)

## iCommands For Windows

### Download

iCommands can be downloaded from downloads page of [www.petashare.org](http://www.petashare.org). After the download it has to be extracted.

### *Setting up the Petashare User Environment file in Windows (for i-commands only)*

- Running i-commands requires setting up an env file, ".irodsEnv", which at least contains the following :

```
irodsHost      ericl.loni.org (can be any of the 7 sites)
irodsPort      1247
irodsUserName  webuser
irodsZone      tempZone
irodsDefResource ericl
```

On a Windows machine, this file should be placed in the following directory (or folder).

```
%HOMEDRIVE%%HOMEPATH%\irods
```

where %HOMEDRIVE% and %HOMEPATH% are two built-in environment variables in Windows. And the %HOMEDRIVE%%HOMEPATH% is the home directory (folder) for a XP/Vista user. When a "Command Prompt" is launched, a user is automatically placed in this place.

The directory (folder) is often like following:

```
C:\Documents and Settings\bzhu
```

where 'bzhu' is the user name.

In this example, the env file has the following full path.

```
C:\Documents and Settings\bzhu\irods\irodsenv
```

To create the env file:

- Launch a "Command Prompt" by navigating to the menu "Start" -> "Accessories" -> "Command Prompt".
- Change directory to the user home directory.
  - cd %HOMEDRIVE%%HOMEPATH%
- Type the following Windows command to create a folder, ".irods", and move into this directory.
  - md .irods
  - cd .irods
  - Notepad .irodsEnv

(This will launch a Notepad and create a text file named ".irodsEnv".)

- Put the irods content into the Notepad and click save.

### *Setting up the PATH for iRODS executables (for i-commands only)*

To run i-commands in any directory in a Windows machine, the path to where i-commands reside should be set in the Windows PATH environment variable.

This is the same concept as the PATH concept in a UNIX machine.

To do this, launch the System dialogue via:

- Start -> settings -> control panel.
- Click the "System" icon.

In the "Advanced" tab, click the "Environment variables" button.

Add the path for i-commands in the "PATH" either in user category or the system category.

Example Commands:

- `iinit.exe` Initialize - Store your password in a scrambled form for automatic use by other icommands.
- `iput.exe` Store a file
- `iget.exe` Get a file