

## Table of Contents

<b>File Systems</b>	<b>2</b>
Available file systems	2
Notes	3

## File Systems

### Available file systems

On the DEEP system, three different groups of file systems are available:

- the [?JSC GPFS file systems](#), provided via [?JUST](#) and mounted on all JSC systems;
- the DEEP parallel BeeGFS file systems, available on all the nodes of the DEEP system;
- the file systems local to each node.

The users home folders are placed on the shared GPFS file systems. With the advent of the new user model at JSC ([?JUMO](#)), the shared file systems are structured as follows:

- **\$HOME**: each JSC user has a folder under `/p/home/jusers/`, in which different home folders are available, one per system he/she has access to. These home folders have a low space quota and are reserved for configuration files, ssh keys, etc.
- **\$PROJECT**: In JUMO, data and computational resources are assigned to projects: users can request access to a project and use the resources associated to it. As a consequence, each user can create folders within each of the projects he/she is part of (with either personal or permissions to share with other project members). For the DEEP-SEA project (for example), the project folder is located under `/p/project/deepsea/`. Here is where the user should place data, and where the old files generated in the home folder before the JUMO transition can be found.

The DEEP system doesn't mount the **\$SCRATCH** file systems from GPFS, as it is expected to provide similar functionalities with its own parallel and local file systems.

The `deepv` login node exposes the same file systems as the compute nodes, but it lacks a local scratch file system. Since `/tmp` is very limited in size on `deepv` please use **\$SCRATCH** instead (pointing to the project folder) or use e.g. the `/pmem/scratch` on the `dp-dam` partition **\$LOCALSCRATCH** on any other compute node when performing SW installation activities. **A quota has been introduced for `/tmp` on `deepv` to avoid clogging of this filesystem on the login node which will lead to several issues. Additionally, files in `/dev/shm`, `/tmp` and `/var/tmp` older than 7 days will be removed regularly** ""

The following table summarizes the characteristics of the file systems available in the DEEP and (SDV) systems. **Please beware that the `$project` (all lowercase) variable used in the table only represents any JuDoor project the user might have access to, and that it is not really exported on the system environment.** For a list of all projects a user belongs to, please refer to the user's [?JuDoor page](#). Alternatively, users can check the projects they are part of with the `jutil` application:

```
$ jutil user projects -o columns
```

System	File System	Mount Point	Access	Quota	Notes
JSC	GPFS	/p	Read/Write	High	Shared across all JSC nodes
DEEP	BeeGFS	/p	Read/Write	High	Parallel file system
DEEP	Local	/	Read/Write	Low	Local to each node
DEEP	Local	/tmp	Read/Write	Very Low	Limited size
DEEP	Local	/var/tmp	Read/Write	Low	Subject to cleanup
DEEP	Local	/dev/shm	Read/Write	Low	Subject to cleanup
DEEP	Local	/pmem/scratch	Read/Write	High	Alternative scratch space
DEEP	Local	/p/project	Read/Write	High	Project-specific data
DEEP	Local	/p/home	Read/Write	High	User home folders

## Notes

- dd test @dp-dam01 of the DCPMM in appdirect mode:

```
[root@dp-dam01 scratch]# dd if=/dev/zero of=./delme bs=4M count=1024 conv=sync
1024+0 records in
1024+0 records out
4294967296 bytes (4.3 GB) copied, 1.94668 s, 2.2 GB/s
```

- The /work file system which is available in the DEEP-EST prototype, is as well reachable from the nodes in the SDV (including KNLs and ml-gpu nodes) but through a slower connection of 1 Gb/s. The file system is therefore not suitable for benchmarking or I/O task intensive jobs from those nodes

For moving data between /p/\* and /arch, please use JUDAC instead of performing these actions on the login node (deepv). This helps avoiding congestion on the Just connection:

```
ssh -l <username> judac
mv /p/... /arch/...
```